GENERIC WEB-BASED ADAPTIVE TUTORING SYSTEM FOR LARGE CLASSROOM TEACHING

HU YINGPING

NATIONAL UNIVERSITY OF SINGAPORE

2009

GENERIC WEB-BASED ADAPTIVE TUTORING SYSTEM FOR LARGE CLASSROOM TEACHING

HU YINGPING
(B.ENG., M.ENG., XJTU)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2009

ACKNOWLEDGEMENTS

First, I would like to express my deep and sincere gratitude to my supervisor Associate Professor Lian Yong for his kind support and valuable guidance throughout the whole process of my research work. Prof Lian's stimulating suggestions and encouragement helped me in all the time of research. His profound knowledge, abundant experiences and the way of conducting research have been of great value for me. Without his understanding, inspiration and guidance I could not have been able to complete this project successfully.

Many thanks should be given to my colleagues in the Signal Processing and VLSI Design Laboratory for their support and joy given to me during these four years.

My deepest appreciation goes to my family for my parents' dedication, love and persistent confidence in me. I own my loving thanks to my husband He Hongpu. Without his encouragement and understanding, it would be impossible for me to finish this work. This thesis is dedicated to all of them.

The financial support of National University of Singapore is greatly acknowledged.

Last but not least, I would like to thank everyone who had helped, in one way or another, towards the completion of this project.

TABLE OF CONTENTS

SUMMARY

Teaching large classes is a very challenging task for educators due to the divers background of students and differences in learning styles. To improve the learning outcomes, it is necessary to explore new ways to facilitate teaching and learning in large class. Intelligent educational tool is one of the candidates, which is able to emulate small class teaching, honor the individual student's uniqueness and provide appropriate tutoring function to achieve better learning outcome.

Intelligent Tutoring Systems (ITSs) and Adaptive Hypermedia Systems (AHSs) are the two main techniques being widely adopted for adaptive or personalized tutoring. ITSs provide adaptive tutoring for each student and decide how, when and what to do next during a tutoring session based on the student model. Although ITS is adaptive in presenting tutorial questions, it does not allow students to freely explore the information space. AHSs, on the other hand, give student full access to all learnt and ready-to-be-learnt materials, it lacks in "intelligence" to make pedagogical decisions. In this research, we propose an Adaptive Tutoring System (ATS) for large class teaching. ATS integrates the student modeling technique in ITS and free access concept in AHS to form a web-based interactive, adaptive and personalized environment. To reduce the workload in constructing ATSs, a Web-based Authoring Environment (WAE) is developed. The combination of the ATS and WAE forms a Generic Web-based Adaptive Tutoring System (GWATS). Our initial experiments show that GWATS significantly reduces the time for constructing ATS and it enhances learning performances in a large class.

Another goal of this research is to develop a prototype system trying to derive the student's motivation states from their learning behaviors, taking motivations into account and using Dynamic Decision Network (DDN) to make pedagogical decisions. For the prototype implementations, we used our best judgment to set default values for Conditional Probabilities Table (CPT) parameters, prior probabilities and utilities. Further works are needed to obtain accurate values of CPT. For the sake of simplicity, the model described in the motivational prototype system covers only the general model, and includes only a subset of the variables that are necessary to derive motivation states. We chose this subset to show how the model is built and how it works, but several additional variables should be included to model real interactions.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AI	Artificial Intelligence
AHS	Adaptive Hypermedia System
ANOVA	Analysis of Variance
ATS	Adaptive Tutoring System
BN	Bayesian Networks
CG	Controlled Group
CPT	Conditional Probability Table
DBN	Dynamic Bayesian Network
DDN	Dynamic Decision Network
FEG	Fully Experimental Group
GWATS	Generic Web-based Adaptive Tutoring System
IES	Intelligent Educational System
ITS	Intelligent Tutoring System
MATS	Motivational-based Adaptive Tutoring System
PEG	Partial Experimental Group
WAE	Web-based Authoring Environment

CHAPTER 1

INTRODUCTION

With increased enrolment and shrinking budgets at colleges and universities, teaching large classes in higher education becomes unavoidable. Over the past two decades, considerable research has been done to promote and develop different teaching mechanisms and various learning platforms for effective teaching and learning, especially in large classes. Personalized instruction is "the effort on the part of a school to take into account individual student's characteristics, needs and flexible instruction practices in organizing the student's learning environment"[1]. Personalized learning is an approach within a learning environment that tailors learning according to individual needs. The intent of personalized learning is to choose appropriate teaching strategies to engage each student in the learning process in order to match their abilities, preferences and motivations. Personalized learning acknowledges individual differences among students, and one of its most important aspects is to identify the underlying differences that influence learning. In large classes filled with students with varying preferences in their approaches to learning, personalized learning seems to be the most effective model for improving learning efficiency.

Intelligent Tutoring System (ITS) [2] and Adaptive Hypermedia System (AHS) [3] are the two main techniques for individualized tutoring, and both are widely acknowledged and accepted by educators. Based on a student's knowledge state obtained from that student's model in ITS or a user model in AHS, these systems automatically diagnose the student's current learning status and personalize the learning environment and instructions to match the student's learning state. These systems facilitate students' learning and take a significant workload off the educators, especially in large classes. Educators can therefore focus on improving their teaching quality rather than performing tedious or complex routine tasks. Although ITS allows "mix-initiative" tutorial interactions where students can ask questions and have more control over their learning, basically it's the ITSs specifies what to teach and how to teach it based on the student model and adapts the instructions to each user. AHS, on the other hand, is a student-centered learning environment based on adaptive presentation and navigation technologies, which allows students to access all learned and ready-to-be-learned materials [7]. This research has developed an adaptive tutoring system (ATS) that combines the benefits of ITS and AHS. The ATS incorporates intelligent tutoring techniques, offers the freedom of explorer learning, dynamically adapts to the individual user's knowledge level and learning goals, provides intelligent guidance and supports the user in acquiring knowledge. The system organizes the learning materials and manages the learning strategies in a learning environment centered on the students. This proposed ATS aims to alleviate some of the problems faced when teaching large classes.

As is well known, knowledge-based ITSs are difficult to construct. Each one must be built from scratch at a significant cost. As a result, the applications of ITS and AHS are limited. So there is an urgent need to develop an easy way to use ITS and AHS that helps educators take advantage of available technologies to enhance learning in schools and universities. In this research, a Web-based authoring environment (WAE) was developed to simplify the construction of affordable and effective adaptive tutoring systems to enhance the teaching and learning efficiency in large classes. A tutoring system based on a WAE represents the knowledge domain as a concept structure and models students with a Bayesian network (BN). Based on the Bayesian student model, the generated tutoring system provides individualized tutoring and instant feedback to each student.

Knowledge states cannot typically represent characteristics that vary from individual to individual. Studies show that, besides individual ability, certain personal characteristics, such as the student's learning style and motivational states, are considered important and play a key role in the teaching and learning process. Learning style is the unique way a person habitually approaches or responds to the learning task [4], which influences the way the student acts toward the learning environment. Besides learning style, emotion is another factor affecting learning. For example, a poor teaching strategy can lead to negative motivation that impairs learning. Students' learning performances improve significantly if the students are provided with appropriate learning materials or methods at certain moments under certain conditions. Highly motivated students usually perform better than less motivated students. Therefore, considering students' learning styles and cognitive characteristics may contribute to increasing the effectiveness of intelligent educational systems, especially for student populations characterized by a wide range of learning abilities, preferences and cognitive profiles. The importance of learning styles and motivations in education has recently caught the attention of many researchers. They attempted to create an individualized learning environment that tailors the teaching strategy to the individual and promotes positive motivation. A prototype of the Motivation-based Adaptive Tutoring System (MATS) was developed in this research. MATS details how to recognize students' motivation states through observable learning behaviors and then reacts accordingly to keep the students motivated.

The thesis is organized as follows. Chapter 1 provides an overview of the thesis. Section 1.1 covers the overall context of this research and presents its objectives and originality. Section 1.2 presents an overview of learning styles and motivational states and their impacts on learning. Section 1.3 reviews of the existing intelligent teaching and learning tools. Section 1.4 summarizes proposed authoring tools. In Section 1.5, the scope, objectives and contributions of this research are listed. Finally, Section 1.6 reflects on the organization of the thesis and suggests future work.

1.1 Teaching Large Classes

Teaching a large class has always been a challenge for educators due to the many difficulties imposed on the teaching-learning process [5]. These include [6]: working with diverse student needs and backgrounds, meeting the needs of all students, giving students instant feedback, engaging students in active learning, keep track of students' learning behaviors, personalizing the learning experience and motivating students. How can teachers overcome these difficulties and enhance the learning experience in a large class? One possible solution is to leverage the vast experience accumulated in teaching small classes. To do so, we need to identify the differences between large and small classes and try to emulate a small-class environment in a large one to achieve better learning outcomes. It is generally accepted that learning outcomes are inversely proportional to class size, i.e., the smaller the class, the more the student learns. However, recent findings revealed that class size does not necessarily correlate to learning outcomes [7]. The size of a class is not the most important factor affecting the learning outcomes; rather, the characteristics of the instructor, the way the course is organized and how it is taught play important roles in the learning process. Therefore, in theory the efficiency of teaching a large class can be as good as that in a small class as long as the teachers have the same good strategies. The main advantages small classes have over large ones are that they provide students with a personalized learning environment, engage students in active learning and give students instant and appropriate feedback. These advantages lead to higher teaching quality and greater student satisfaction [8].

To duplicate a small class environment in a large one without incurring additional labor costs, many researchers [9-13] have proposed different ways to address issues in a large class, especially in engineering education. It seems that the most effective model is individual tutoring or personalized tutoring [14]. Personalized tutoring honors and recognizes the unique gifts, skills, needs and interests of each student and then tailors the tutoring to the uniqueness of each individual. The key to improving learning efficiency in large classes is to acknowledge and identify the differences among students. Creating a personalized environment tailored to the students' different needs is the solution to facilitate better learning in large classes. With the rapid growth of Internet access to the World Wide Web, many researchers have acknowledged the numerous advantages of web-based education systems: 1) convenient accessibility that lets students learn at their own pace from anywhere at any time, 2) compatibility and interoperability among different platforms that allow easy incorporation and interoperable contents and services, 3) efficient communication and wide coverage of the Internet for flexible and effective channels of online communication among teachers and students and 4) educator support [15]. These advantages can potentially bring the individual tutoring experience to a large class and provide an individualized learning environment for each student without incurring much additional cost.

In response to the pressures and challenges of teaching a large class, the uniqueness and the huge cost of personalized learning, along with the potential advantages of web-based education, it is important to develop a web-based personalized learning environment that provides teachers and students with tools for after-class teaching and learning activities.

1.2 Learning Styles and Motivational States

The first challenge of personalized learning is to identify the individual differences among students. It is a well-known fact that, despite the individual's knowledge state, how a student perceives, gathers and processes material and his or her emotions or motivations all play a key role in teaching and learning [16]. Positive motivation contributes to learning achievement, while negative motivation has the opposite affect [17, 18]. Hence, it is crucial for intelligent education systems to adaptively treat the students' distinctive information such as interests, learning styles and motivation [19-21].

"Learning style" denotes the typical ways in which a student takes in and processes information, makes decisions and forms values. Each individual has his or her own way of learning. A person's learning style is reflected in his or her behavior, and it can greatly affect his or her learning outcomes [22, 23]. One instructional environment cannot possibly fit all students [24], because students have different learning styles as they take in and process information [25]. They might learn more effectively when the instruction is matched to their individual learning style [26].

Much research has been carried out on learning styles. Meanwhile, many learning style theories have been established. The most widely used are Kolb's Learning Style

Theory [27], Gardner's Multiple Intelligence Theory [28] and Felder-Silverman Learning Style Theory [29, 30]. In recent years, the importance of modeling and using learning styles has been widely acknowledged. Many researchers have started to consider learning styles in computer-based educational systems. Lots of systems have been built to take care of students' learning style [31-36]. A large class usually consists of a wide spectrum of students differing from each other not only in race, culture, age and background, but also in personal traits (e.g., intelligence), self-confidence, motivation and the preferred type of learning methods and learning styles. It is important to address these distinct characteristics.

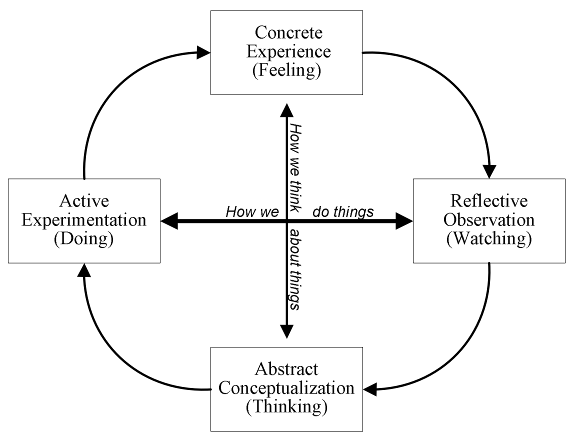


Figure 1-1: Kolb's Learning Cycle

According to Kolb [37], there are four sequential stages in the learning cycle (Figure 1-1). Concrete Experience provides a basis for experiences and is followed by Reflection and Observation on that experience. Reflection and Observation are then assimilated and distilled into Abstract Conceptualization that produces new implications for action, which can be termed Active Experimentation. Based on Kolb, Honey and Mumford [38] suggested four types of learners: Activist, Pragmatist, Reflector and Theorist. An Activist prefers doing and experiencing; a Reflector observes and reflects; a Theorist wants to understand underlying reasons, concepts and relationships; and a Pragmatist likes to “have a go” and try things to see if they work.

With the increasing diversity in the student population, the ideal learning environment for large classes should include all of the four types. Students are encouraged to start with his or her favorite learning activities, then continue with less “style-matched” activities to develop new capabilities [4]. Meanwhile, the student’s preferences on learning materials or leaning activities might change over time within various circumstances. Instead of detecting the student’s dynamic learning preferences and then tailoring to those, we chose to provide multiple types of learning material to fulfill each student’s preferred learning style and his/her dynamic preferences. Within the proposed learning environment, the student is free to choose learning material and is encouraged to learn throughout the learning cycle.

Motivation is another key element of education and plays a crucial role in students’ success. Weiner [39] defines motivation as “the study of the determinants of thought and action—it addresses why behavior is initiated, persists and stops, as well as why choices are made.” From this definition, we can derive that motivation motivates helps students to learn, affects the quality of the efforts they invest and influences the choices they make. Meanwhile, motivation of the student might be affected by tutoring and the learning environment. However, most intelligent education systems have overlooked the motivational aspects of learning. There are two main concerns about tailoring to motivation aspects: how to detect students’ motivational states and how to respond to keep them motivated, especially for web-based learning. This thesis presents a prototype of the Motivation-based Adaptive Tutoring System (MATS), which details how to recognize students' motivational states through observable learning behaviors, then reacts accordingly to keep the students motivated.

1.3 Intelligent Education System

Personalized learning advocates that learning should not be restricted by time, place or other barriers, but should be tailored to the continuously changing individual student's background, requirements, abilities and preferences [40]. Of all the interesting methods and techniques used to provide adaptation or personalization, ITS and AHS are the two main techniques most widely adopted.

An ITS is a computer-based educational program that provides direct customized instruction and personalized feedback to students. Most ITSs are based on Artificial Intelligence (AI) techniques [41] and are generally known for their abilities to identify a student's learning state and replicate the process of one-on-one instruction in a small classroom. The intelligence of ITS comes from the information related to a student's knowledge, the specific domain knowledge, the teaching strategies and the learning environment, which are represented by the four basic components in ITS, i.e., the domain model, the student model, the tutoring model and the user interface. The domain model contains the information to be taught, the source of the knowledge and the standards for evaluating the student's performance. The existing student model stores a description of a student's knowledge and learning traits, which generally falls into two categories: the domain-specific information, such as the student's current knowledge state relating to a specific domain [42], and the domain-independent information, such as the student's learning profile, his or her learning style and his or her current motivational state. The student model in this research will follow this convention and split student characteristics into two categories: knowledge-related information in the knowledge model and domain-independent information in the psychological model. The details of the student model will be presented in Chapter 3. The tutoring model makes pedagogical decisions and decides what, when and how to teach based on the domain and the student models. The fourth component of ITS is the user interface or learning environment, which offers a friendly channel for the student to communicate or interact with ITS. From the user's point of view, most of ITS can be considered as a user interface [43], which highlights the importance of the user interface to ITS. Based on the four components, the ITS can simulate a human tutor by putting their knowledge and inference mechanisms into a computer system, make inferences about the student's knowledge based on the student's response, instantly provide adaptive feedback, intelligently decide the next best pedagogical action and deliver adaptive instruction. The ultimate goal of ITS is to provide a personalized learning environment. Evaluations reveal that ITS is highly effective compared with traditional instructional methods, thanks to the built-in intelligence that helps to identify students' needs and provides highly individualized tutoring through curriculum sequencing, intelligent diagnosis of a student's answers and interactive feedback and support [44].

With the rapid development and deployment of Internet, AHS is a relatively new research area in contrast to the traditional "one-size-fits-all" approach of standard online learning. According to Peter Brusilovsky [55], AHS builds a model of the goals, preferences and knowledge of each user, then use this model to personalize the content and hypermedia pages for each individual. Unlike the ITS's direct tutoring guidance, AHS adopts adaptive navigation support technology on the link level to support the student in hyperspace orientation and navigation. The adaptive presentation on the content level adapts the content of a hypermedia page to meet the individual's needs based on his or her user model [44]. AHS enables students access to all learned and ready-to-be-learned materials and provides a student-centered learning environment [45]. However, without direct guidance, it is easy to get lost in hyperspace.

1.4 Authoring Tools

Intelligent Educational Systems (IES), including ITS and AHS, are well known for personalized tutoring [46]. Evaluations reveal that IES is highly effective compared with traditional instructional methods by intelligently identifying students' needs and providing highly individualized tutoring [47-51]. However, an IES is rarely used in real educational situations. The underlying reason might be that the IES has to be built from scratch at a significant cost. The estimated effort used for development time varied from 200-300 hours of authoring for one hour of instruction [52, 53]. Besides, most IESs are created for a specific domain, and it is difficult to reuse them in other domains without much time and effort. The difficulty and complexity of creating an IES motivates the development of authoring tools to simplify construction and create cost-effective IESs, which might promote IESs into wider applications.

1.5 Research Objectives and Contributions

From the above discussion, it is clear that most of available teaching and learning tools do not satisfy the requirements of large classes. This prompts us to develop an intelligent educational tool for assisting educators in large classes. The purpose of this thesis is to develop teaching and learning assistance tools to improve learning efficiency in large classes. The primary goals include the following:

- To design an ATS that provides a personalized learning environment to cater to individual needs, which is the integration of traditional ITS and AHS. Since an ATS is not as effective as a human tutor and it is impossible to replace such a tutor, it is best used as a supplementary, after-class tutorial tool. Students still need to attend classes given by human teachers.
- To ease construction and promote ATSs into wider applications, the proposed Generic Web-based Adaptive Tutoring System (GWATS), including a web-based authoring environment (WAE) as part of its components, enables effective construction an ATS. All ATSs constructed by GWATS use the same tutoring model and share the generic adaptive tutoring strategies.
- To evaluate the effectiveness of the generated ATS and the generic tutoring strategies.
- To develop a prototype MATS, taking students' motivational states into account when responding to a student in order to keep him or her motivated.

The main contribution of this work can be summarized as follows:

1. GWATS integrates the student modeling technique in ITS and free access concept in AHS to form a web-based interactive, adaptive and personalized environment. GWATS maintains the domain and student model and dynamically tailor the instruction to the specific needs of the student. Meanwhile, the tutoring model incorporates the features of AHS, i.e., sharing control of instructions with the student and allows students to freely browse the learning environment at a certain level. This contribution takes the advantage of both systems and improves students' learning performance.

2. The novel architecture of GWATS integrates the authoring components to the standard ITS system structure to form GWATS. This contribution decreases the effort and the skill threshold in constructing ATS. The web-based characteristics of GWATS allow instructors to construct ATS and deliver them over the WWW, which makes teaching a large class and distance education more convenient.
3. Bayesian network is employed in the authoring environment. This contribution provides a novel way to define the domain structure and to model the independency relationships between different learning units. Domain knowledge representation is the "heart" of the intelligent tutoring system. This makes a contribution by accurately model the domain knowledge.
4. The prototype system based on the behavior tracking and analysis module and DDN technology reveals the working mechanism of how to infer the motivational states through observable learning behaviors and how to respond to the detected motivations to keep students highly motivated. Although the efficacy of the system will be further investigated by real students, the architecture of GWATS with behavior tracking and analysis model and the studies carried out in Chapter 6 gave us strong confidence on the performance of the motivational tutoring system.

1.6 List of Publications

[1] Y.P. Hu, X.Q. Zhang, J.H. Yu and Y. Lian, "An online learning module for advanced digital filter design techniques," in Proceedings of the Global Conference on Excellence in Education and Training 2004, Singapore, 2004.

[2] Y. Lian and Y.P. Hu, “An automatic grading system for adaptive teaching,” in Proceedings of the Global Conference on Excellence in Education and Training 2004, Singapore, 2004.

[3] Y. Lian and Y.P. Hu, “An Integrated e-Learning Platform for Learning-by-Doing in Large Classes,” in Proceedings of International Conference on Engineering Education University of Florida, , Florida, USA, October 17-21 2004.

[4] Y. Lian and Y.P. Hu, “Enabling Adaptive Teaching in a Large Class,” In TLHE 2004: International Conference on Teaching and Learning in Higher Education, Singapore, 1-3 December 2004.

[5] Y. Lian and Y.P. Hu, “Enabling learning-by-doing in a large class with the help of an e-learning platform”, INNOVATIONS 2005: World Innovations in Engineering Education and Research, pp.123-134, 2005.

[6] Y.P. Hu and Y. Lian, “An Open Learning Environment for Large Classroom Teaching,” International Conference IEEE ICL'2006 Interactive Computer Aided Learning, Villach, Austria, 27-29 September 2006.

[7] S.W.K. Wah, Yingping Hu and Yong Lian, “LearningVista—A Learner-centered e-learning Platform,” In TLHE 2006: International Conference on Teaching and Learning in Higher Education, Singapore, 6-8 December 2006.

[8] Y.P. Hu, Q. J. Chong and Y. Lian, “Web-based Adaptive Tutoring System,” In TLHE 2006: International Conference on Teaching and Learning in Higher Education, Singapore, 6-8 December 2006.

[9] Y.P. Hu and Y. Lian, “An Adaptive E-learning Portal for DSP Learning,” In ICICS07 2007: Sixth International Conference on Information, Communications and Signal Processing, Singapore, 10-13 December 2007.

[10] Y.P. Hu and Y. Lian, "Web-based Authoring Environment for Building Adaptive Tutoring System", IEEE Transaction on Education, under preparation.

[11] Y.P. Hu and Y. Lian, "Developing web-based adaptive tutoring system with the authoring environment WAE", IEEE Transaction on Education, under preparation.

1.7 Organization of Thesis

The rest of this thesis is organized as follows:

Chapter 2: This chapter gives a literature review of the previous works related to adaptive tutoring system, Bayesian network-based student modeling, the factors influencing student learning in large classes and the existing authoring tools.

Chapter 3: This chapter presents the GWATS design. The system's architecture is presented first, followed by the functions of each component. The generic tutoring model, running in the backend, is applicable to all ATSs constructed by WAE. The details of the generic tutoring model and tutoring strategies are presented in this chapter.

Chapter 4: Details of the WAE are elaborated in this chapter. WAE is a key component of GWATS, which enables the efficient construction of an ATS by using the universal hierarchical domain structure, the same student model framework and the generic tutoring model. Design considerations are discussed first, followed by the details of the authoring environment and authoring process.

Chapter 5: This chapter performs a series of experiments to evaluate the effectiveness of the ATS constructed by WAE. The experimental results validate the effectiveness of the Bayesian student model and adaptive tutoring policies. The overall performance of the GWATS is revealed based on the survey results.

Chapter 6: This chapter develops a prototype MATS to infer students' motivation states, like confidence, independence and effort, based on Del Soldato and Du Boulay's motivational planning approach. The prototype was based on the GWATS architecture. The behavioral analysis module uses Bayesian modeling techniques and considers knowledge and motivational states in making pedagogical decisions. It focuses on how to react to the detected motivational states, and how to keep students in the optimal emotional states, instead of how to detect students' motivation.

Chapter 7: The thesis concludes by showing how the goals of this project have been met, the important results and future work.

CHAPTER 2

REVIEW OF EXISTING TEACHING AND LEARNING TOOLS

This research attempts to develop a web-based adaptive tutoring system for improving the effectiveness and efficiency of teaching and learning in large classes. It is important to understand the existing techniques and how these techniques help to enhance the learning experiences in large classes. In the following sections, we will briefly review the past research and highlight the components needed for an efficient learning tool.

2.1 Adaptive Tutoring System (ATS): Integration an Intelligent Tutoring System with Adaptive Hypermedia System

ITS and AHS are regarded as two different web-based approaches on education. These approaches are, in fact, complementary.

ITSs are computer-based intelligent instructional systems. They provide customized instruction and personalized feedback to students. The goal of an ITS is to function as a human tutor and provide an individualized one-on-one environment based on the knowledge about domain, the beliefs of the student and the teaching strategies, but without the expense of having a human tutor. ITSs have proven to be effective at increasing learning performance compared with traditional methods and have significantly improved learning outcomes [54]. Most of ITSs are controlled by the embedded tutoring strategy to decide how, when and what to do next based on the student model. The ITS usually does not allow students to freely explore the information space. Such restrictions affect the efficiency of learning, especially in large classes. AHS, on the other hand, is a student-centered learning environment based on adaptive presentations and adaptive navigation technologies, which allow students to access all learned and ready-to-learn materials. However, an AHS lacks control of the learning process. Without such control, the student can easily get lost in space, work inefficiently and face difficulties in discovering some important features of the subject.

AHSs maintain a user model containing personal information, then use this model to adapt to the individual needs throughout an interaction process [55]. The main goal of AHSs is to provide personalized views of hypermedia responding to different goals, preferences, interests and knowledge of the student based on adaptive presentations and adaptive navigation technologies [56]. Adaptive presentation technology adapts the content to the user model. Pages presented to students in a system with adaptive presentation are not static but are adaptively generated for each individual. Adaptive navigation supporting technology assists navigation by limiting browsing space, suggesting the most relevant links to follow or providing adaptive comments to visible links. To sum up, AHSs have demonstrated their potential to offer students freedom of browsing course materials while ensuring that the materials are always relevant and matched to the students' levels. However, the user model in an AHS is insufficient, and it is difficult to measure the knowledge that a student gains in AHSs. While AHSs give students the freedom to access all the learning materials presented, the adaptivity can make system much less usable if the users do not understand how the resources are organized and, consequently, they can easily get lost in the space [57].

Therefore, an AHS needs to be supplemented by explicit tutoring and guidance [58]. This guidance is an important ingredient of effective learning, and an ITS can provide this ingredient. Meanwhile, the hypermedia approach in an AHS can add a new dimension to an ITS by providing freedom for students' exploration and acquisition of domain knowledge.

In this research, we aim to develop a web-based ATS that combines the benefits of ITS and AHS and provides an interactive, adaptive and personalized learning environment. The system controls the organizing of the learning materials and manages the learning strategies, while the learning environment centers on the learners. The system enables students to actively participate in a self-directed learning process, allows students to take charge of his or her own learning pace and actions and provides mechanisms for adjusting the learning program to match the learning attributes. The parameters that most frequently govern adaptivity in existing ITSs and AHSs are the student's existing knowledge and skills. However, most systems neglect the students' cognitive and motivation characteristics.

2.2 Learning Styles Consideration

ITSs and AHSs are capable of providing individualized instruction like a human tutor by deciding how, when and what to teach based on the students' knowledge states. However, individuals differ from each other in many aspects, for example, learning abilities or knowledge states, such as cognitive, affective and social-cultural characteristics [59]. These individual differences are fundamental to learning and the students' characteristics should be emphasized in a learning environment. It is expected that enhancement of the effectiveness can be achieved by recognizing and responding to students' learning needs, their diversification of learning styles and their preferences. Therefore, intelligent educational systems need to take the cognitive and motivational traits of the students into account and provide them with adequate responses from pedagogical, cognitive and motivational points of view.

Research suggests that for students with various learning styles, it is better to apply teaching styles that match their learning styles. Identification of the learning style of each individual is a prerequisite for learning styles. Over the years, a number of researchers have come up with various strategies for defining and categorizing the learning styles of individuals [60-63]. However, all of these strategies rely on the individual subjectively responding to a series of questionnaire items. It is not apparent whether individuals are able to describe or conceptualize their own learning processes. If the questions are too long or students are not aware of the consequences or usage of the questionnaires, they tend to choose answers arbitrarily. Therefore, measuring learning styles using pre-designed instruments might result in an inaccurately extracted style. Alternatively, there are style-matching strategies using AI technologies such as the Bayesian network [64] or neural network [65] to identify students' learning styles [66].

Style-matching strategy is frequently employed to adapt the instructional style to match the students' identified learning style and to improve leaning performance by matching learning style with instructional presentation [67-71]. The individual learning style is diagnosed once and will be used as a benchmark and kept static to provide individualized tutoring later. This is based on an assumption that learning style has temporal stability and an individual's learning style remains relatively constant across a period of time. This, however, has not been proven by research to date [72].

Instead of identifying a learning style once for each individual, then providing adaptive instruction and a strategy to match that style, we have proposed a web-based ATS, which treats learning styles as a dynamic component and provides several types of learning materials and methods for individuals and caters to students with various and changing learning styles.

2.3 Motivational States Consideration

As Goleman [73] reminds us, "The extent to which emotional upsets can interfere with mental life is no news to teachers. Students who are anxious, angry, or depressed don't learn; people who are caught in these states do not take in information efficiently or deal with it well." Therefore, one of the main concerns in education is to consider students' motivational states and keep them engaged in learning. Human tutors can detect the students' emotional states and variations and can devote as much time to achieve students' motivational goals as to cognitive and information goals [74]. For computer-based tutors, there is some research about attempting to motivate students by using interactive digital media [75]. However, this approach can only increase students' curiosity, foster their interests and motivate them by showing how to apply their knowledge to the real applications and to understand the underlying principles of their knowledge, thereby contributing to greater engagement, but the approach cannot spark the students' internal motivation.

Detecting the students' motivational state is a crucial step in creating a successful ATS, which incorporates motivation factors. Unfortunately, there is no straightforward way to do this. Various approaches, including questionnaires, verbal communication, self-reports, expert systems and affective computing have been used in computer-based motivation diagnosis [76- 84].

Each method has its own pros and cons. But all of these methods focus on motivation diagnoses without mentioning how to adapt the instruction to the detected motivation. In this thesis, we provide a framework to collect student leaning behaviors initiated during the interaction to diagnose the students' motivational state and how to adapt instructions to this state. Instead of accurately assesses the students' motivation, we focused on how to respond to the detected motivational states and to show whether the inclusion of motivational states benefit students using ATS.

2.4 Student Action Tracking

Educational research shows that monitoring students' learning is an essential component of high-quality education and is one of the major factors differentiating effective schools and teachers from ineffective ones [85]. In face-to-face classroom lectures, the teacher can monitor students' behavior, observe what students say and do, monitor their learning progress to identify gaps in their knowledge and adapt the teaching to the students' comprehension. However, because of the nature of computer-mediated communications, computer-based tutors cannot monitor the students. It is very hard to get specific information about interactions, such as students' understanding of the materials presented, responses to questions and problems and so forth. All of the above information provides teachers with deep insight into the students and enables immediate feedback/reinforcement regarding their learning and their on-task behaviors.

Given the diversities of the students in large classes, it is crucial for an ATS to distinguish the individual from others and personalize the interaction. A number of existing web-based tutoring systems provide adaptations to various types of users [86] [87]. The general principle behind these adaptations is the stereotype model, which uses an initial interview or questionnaire to gather information, then classifies users into categories. These systems match each student's profiles to one of a number of pre-defined system user profiles. This technique simplifies system design, but the accuracy of matching a stereotypical user with the needs of an actual user is questionable [88]. Besides, the student's stereotype might change during a session. To maintain an appropriate and powerful student model, the mechanism for monitoring the student interacting with the system and updating a student model dynamically and accordingly is need.

Learning style, motivational states and other hidden characteristics can be derived by monitoring the interaction with the system and the student's observable behaviors [89]. In light of the perceived needs, we developed a behavior-tracking component for students' initiated actions within the interaction of the ATS. Monitoring how students behaved in the online learning environment enriched the knowledge about their dynamic characteristics and real-time needs. This, in turn, allows better system adaptation based on their recent behaviors [90], enables the development of personalization strategies and helps to increase the system's performance. Meanwhile, monitoring students' usage of the tutoring system supports the evaluation of the system against its initial specifications and objectives.

2.5 Student Modeling Using Bayesian Networks

The student model is a key component in traditional ITS, representing conceptual knowledge of the student to infer the degree of that student mastering the domain knowledge. Various AI techniques have been used to represent student models. Fuzzy logic techniques have been used [91] to handle the inherent uncertainty of a student's behavior and to achieve a description of that student's knowledge. Neural networks have also been used in modeling student attributes to their pattern recognition ability of imprecise or incomplete data, their ability to generalize and learn from specific examples, their ability to be updated quickly with extra parameters and their execution speed [92, 93]. Hybrid neuron-fuzzy synergism has been used for student modeling [94, 95] in which Fuzzy Logic is used to provide human-like approximate diagnoses of students' knowledge, and neural networks are trained to imitate real teachers' tutoring decisions regarding students' characteristics. These approaches did have some success in adaptive instruction, but they required historical data to train the network to work proficiently.

One of the key elements that distinguishes an ITS from a traditional educational system is its ability to interpret student actions by maintaining a model of student reasoning and learning (the student model) [96] and allows the ITS to adapt the interaction to the user's specific needs, as does the user model in AHSs. However, the description of a student model is imprecise and vague, which adds a great deal of uncertainty. Moreover, inferring a student's mastery state from what the system knows and observes entails uncertainty. In addition, uncertainty accumulates in chained inference [97]. The uncertainty of the student's domain knowledge affects the inference or diagnosis of the student's knowledge state, which, in turn, influences the tutoring actions for that student. Therefore, the student model must be theoretically sound enough to deal with all the uncertainty it might encounter.

2.5.1 Basic Probabilistic Knowledge

A Bayesian network is a data structure with great power to represent causal relationships and infer probabilistic outcomes in a domain. Since a student's knowledge is full of uncertainty and characterized by causal relationships and hierarchical structures [149, 150], Bayesian networks are increasingly popular in designing and implementing student models. The definitions about Bayesian networks and basic probabilistic concepts can be found in Russell and Norvig [100] and Nilsson [151]. Since it is impossible to exhaustively examine specific events, when probability theory is used to model the real world, the probability is about a belief in an event based on the observed occurrences thus far. Therefore, probabilities will change after more evidence is available. Before the acquisition of any evidence, the probability can be set to any value or be obtained from small sample data. This probability is called a prior probability and will be refined with more evidence. After the acquisition of new evidence, the updated probability is called posterior probability.

2.5.2 Bayesian networks

Random variables in a domain may have causal relationships. A Bayesian network explains the relationship between independent and dependent variables probabilistically. Technically, a Bayesian network is a directed acyclic graph (DAG) that consists of nodes and links. Each node represents a random variable in a domain, and each link is an arrow that represents a causal influence and points from a node of cause to a node of effect. The parameters used to represent the uncertainty are the conditional probabilities of each node, given each combination of the states of its parents, that is, if $\{X_i, i = 1,...,n\}$ are the variables of the network and $pa(X_i)$ represents the set of the parents of X_i, for each $i = 1,...,n$, then the parameters of the network are $\{P(X_i \mid pa(X_i)), i = 1,...,n\}$. That is, this is the set of discrete conditional probability distributions of each variable, given its parents. This set of probabilities defines the joint probability distribution for the network as,

$$P(X_1,...X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)) \qquad (2\text{-}1)$$

Thus, to define a BN, we have to specify:

1. The set of variables, $X_1, X_2, ...X_N$.
2. The set of links between those variables; these links represent a causal influence between the variables.
3. For each variable X_i, its probability is conditioned to its parents, that is , $P(X_i \mid pa(X_i)), i = 1,...,n.$

In student modeling, each node in the network represents either a concept or a question and their different states. Links between concepts and question shows relationships. A course will consist of several concepts. A concept can be in three states: mastery, partial mastery or non-mastery. A course will also have questions that belong to one or more concepts. A question can be in two states: true or false, where true is when a student has answered the question correctly and false when a student has answered incorrectly.

A Conditional probability table (CPT) is a table that has one probability for each possible combination of parent and child states. As a Bayesian network is a complete model for the variables and their relationships, it can be used to answer probabilistic queries about them. Figure 2-1 shows a simple Bayesian network of three concepts and two questions with their CPTs.

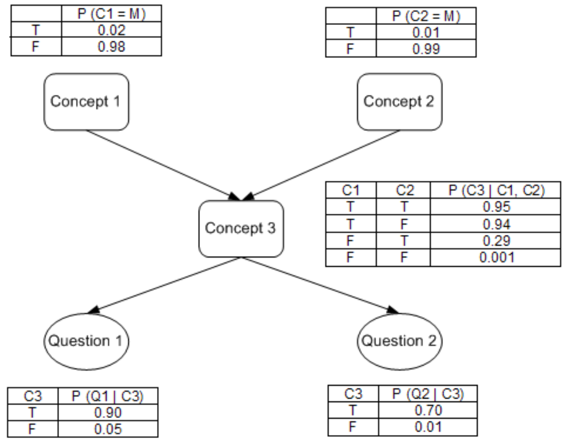


Figure 2-1: Example of a Bayesian network

For simplicity, each node in this network has only two states, true or false. For a concept node, the state true indicates that the concept was mastered and indicates for a question node that the question was correctly answered. This model can be used to answer questions such as "What is the likelihood that a concept 3 is mastered given that a question 1 is correctly answered and a question 2 is incorrectly answered?" These unknown probabilities can be calculated using Baye's Theorem given in the equation shown below.

$$P(A \mid B) = \frac{P(B \mid A)\, p(A)}{P(B)} \tag{2-2}$$

In this equation P (A) and P (B) are called the prior probabilities, as they don't take into account any information about B. P (A|B) is called the conditional probability or the posterior probability because it is derived from the value of B. P (B|A) is the conditional probability of B given A. This process of computing the posterior probabilities of variables given the evidence is called the probabilistic inference.

In the context of ITSs, Bayesian networks have been applied to user modeling (VanLehn et al. 1998) in a diagnostic perspective: Given a student action (symptom), the network provides the most likely state of knowledge (diagnosis). In our work, Bayesian Networks are used to model the user, the structure of knowledge and pedagogical options. The values related to the student's status are taken from the user's interaction with the tutor.

If the student answered the question correctly, then we considered the concept known. Similarly, if the student answered the question incorrectly, then we considered the concept unknown (not known). The probability of each concept being known, namely, $p(a_i = known)$, can then be determined. Moreover, we can also compute $p(a_i=known, P_i=known)$, i.e., the probability that the student correctly answers both the concept a_i and the prerequisite concepts P_i. From $p(a_i=known, P_i=known)$, the desired CPD $p(a_i=known \mid P_i=known)$ can be obtained. Thus, we can calculate every CPD for the Bayesian Network.

Bayesian networks are one of the most effective ways to represent and handle the inherent uncertainties in student modeling. In recent years, Bayesian networks have been utilized in various ways to achieve adaptability in educational systems regarding student knowledge assessments, predications of student goals, determinations of appropriate learning strategies and curriculum sequencing, etc. These applications have demonstrated that Bayesian networks are suitable for effective modeling student behaviors [90]. ANDES (An Intelligent Tutoring System for Physics) teaches Newtonian physics problem-solving techniques to college students and evolved from OLAE (Online Assessment of Expertise) and POLA (Probabilistic Online Assessment) [101]. ANDES uses Bayesian networks to identify the current problem-solving approach of the student. Bayesian networks employed in ANDES [99, 100] relate a student's observable behavior to a particular piece of his or her knowledge. ANDES also uses Bayesian networks for long-term knowledge assessment, plan recognition and prediction of students' actions during problem solving. Bayesian networks in ANDES are constructed automatically from the solution graph associated with each problem. They are domain- specific and used to determine which rules the student has probably mastered according to the observable data (i.e., the student's answers to the questions). Therefore, it is time-consuming and difficult to create Bayesian networks for the domain, which might have thousands of questions. Besides, the generated Bayesian student model is not reusable for other domains. To tackle this problem, we proposed a Bayesian student model allowing the measurement of a student's knowledge at different levels of granularity. Moving the attention from the question level up to the concept level will save time and energy in constructing a Bayesian student model and reasoning on the concept level of granularity. At least the number of concepts each subject contains is remarkably smaller than the number of questions.

2.6 Authoring Tools Review

In the past decade, interest has increased on the use of specialized tools for ITS development, and many authoring tools have been built for ITSs [102-104]. WebCT [105] provides a large variety of support services to students and teachers, but it lacks adaptability. It is more like a learning management system than an authoring system. REDEEM [106] is another well-known authoring tool that allows the teacher to create pedagogical online courses by describing the structure and flow of the content of the course and the sequencing of the content. This tool allows the teacher to divide the course into sections and describe the content that the course will use. REDEEM has been successful in constructing courses. However, it does not provide any adaptivity or dynamic personalization.

InterBook [107] is a tool for authoring and delivering adaptive textbooks on the web To provide adaptivity, this tool relies on the prerequisite relationships among concepts of the domain model and the stereotyped overlay student model. However, the observations about the student's performance and the assessment of mastery of the domain knowledge are mainly based on if a student has read a page or if a student has successfully performed a test related to the concept. This observation cannot provide sufficiently accurate information for individualized tutoring.

WEAR [108] is a web-based authoring tool for adaptive educational systems mainly used for algebra-related domains. WEAR performs student modeling for domain-specific errors as well as algebra-related errors. The domain model containing knowledge about the subject matter is structured as a network of hierarchically organized topics based on the "is prerequisite of" and "is related to" relationships. The student model used for adaptive navigation is a combination of a stereotyped and an overlay student model and stores two attribute-value pairs to represent the estimation of the student's knowledge level on each topic.

The student model keeps track of a student's progress, stores individualized information about the knowledge states of domain knowledge and is an essential component in individualized learning. Various AI modeling techniques have been employed in ITS student models to increase the accuracy of evaluating and predicting students' performance and to improve the system's accuracy. However, none of these authoring tools have incorporated AI modeling techniques yet. With this in mind, we present WAE, a web-based authoring environment, and have built a web-based ATS on a Bayesian student model. Integrating a Bayesian network into an authoring environment by authoring a Bayesian student model and a web-based platform are the distinguishing characteristics of our WAE. Authoring is now more convenient. The other special characteristic of WAE is that it is not a standalone program, but is included as a component in an ATS, which enables the ATS to be applicable to many different domains.

In summary, an effective teaching and learning tool for a large class should contain the following components: representation and structure of the subject knowledge; modeling and maintenance of students' information; an interface handling the interactions between the user and the system; processing engine determining what, when and how to interact with the student; and tools for constructing the educational system. Since the objectives of this thesis were to build a web-based ATS to cope with the difficulties in teaching large classes and provide personalized tutoring for each student, we will demonstrate in the following chapters how to include the above components into an ATS in order to build a successful teaching and learning tool.

CHAPTER 3

GWATS SYSTEM ARCHITECTURE

The Generic Web-based Adaptive Tutoring System (GWATS) system proposed in this study is an adaptive web-based learning environment with a build-in authoring environment applicable to different engineering subjects. The underlying generic tutoring model runs well in the backend functions in different domains. It provides adaptive tutoring to the individual based on the student model, monitors students' learning process and takes into consideration the students' characteristics. This chapter covers the design considerations of GWATS, its architecture, the function of each component and the generic tutoring model.

3.1 Design Consideration

As presented in Chapter 2, ATS is our solution to teaching large classes. It integrates ITS with AHS technologies and uses all the four key components of a traditional ITS: domain model, student modeling, tutoring model and the user interface for interaction with the system. The domain model stores the specific domain knowledge, and the student model monitors the characteristics of each student. The tutoring model provides teaching strategies, while the user interface facilitates interaction between users and the system. In a traditional ITS, domain and student models are usually inflexible, i.e., it is difficult to extend or reuse the existing domain and student models across domains. There are some authoring tools developed for lecturers to create their own systems, but all of these are standalone programs. Once the generated educational system is put into use, it is very inconvenient to modify or add new learning materials or functions.

To create web-based ATSs accessible at anytime, anywhere and editable at anytime, it is essential to integrate a WAE into an ATS to form a GWATS. The function of an ATS is to provide an interactive, personalized learning environment to promote students' learning outcome in a large class. It gives students navigational freedom and provides adaptive tutoring to match their unique needs. It presents multiple types of learning methods and materials to meet each individual learning style and encourages students to learn through Kolb's learning-by-doing cycle. The purpose of the WAE is to simplify ATS construction and promote ATS into wider applications. Integration of the WAE into the GWATS enables lecturers to construct a web-based learning environment for different subjects by creating domain-knowledge contents, loading them into a domain model, compiling the concept-network into the student Bayesian network and deploying the generated ATS through the Internet to interact with students. GWATS is then an ATS with an authoring environment, which makes it applicable to different subjects.

The system architecture of GWATS, its components and the generic tutoring model are presented in the following sections.

3.2 System Architecture

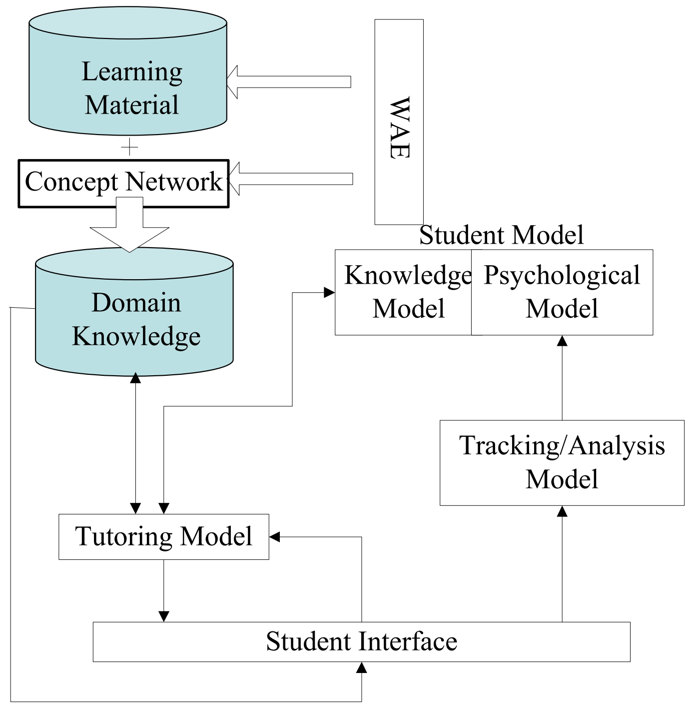


Figure 3-1: GWATS architecture

We proposed GWATS as the solution to simplify ATS creation, to provide adaptive tutoring to individual students, to monitor and collect the learning behavior during the interaction process, to use the students' characteristics in all aspects, to tackle difficulties in large classes and to promote teaching and learning in large classes. GWATS addresses the above issues by (1) using the same hierarchical structure with multiple granularities to represent various domains models and domains knowledge; (2) using a student model, including both a domain-related knowledge model and domain-independent model, to take the knowledge states and the psychological characteristics of the students into account; (3) using a Bayesian student model built on an abstract level to ease the ATS construction and to allow measurement of a student's knowledge at different levels of granularity; (4) using authoring tools to load domain knowledge into domain model and initialize the student model; (5) using a behavioral tracking model to monitor and record the interesting activities initiated by the student during his or her interaction with the system; and (6) using a behavioral analysis model to extract the insight knowledge or characteristics of the individual to facilitate adaptation and reinforce personalized learning.

The architecture of GWATS is shown in Figure 3-1. It is built with a collection of several models important for constructing a learner-centered environment. This system architecture differs from the traditional ITS because of the inclusion of an authoring environment, behavioral tracking and analysis components. Together with the traditional ITS components, these models are connected to form a backend engine to keep the student model updated and provide adaptive, individualized tutoring to each individual based on the student model. The individualized tailoring is realized through the cooperation and communication of all modules. WAE enables lecturers to create a rapid and cost-effective ATS for certain groups of students and on a specific subject by loading domain-related knowledge into the domain model and putting individual student's information into the student model. The student interface is typically an interactive learning environment. Once a student logs in at first time, he or she is presented with the customized learning material. The behavioral tracking module is then kicked in to track the student-initiated actions through the student interface. The analysis module interprets the tracked activities and extracts the students' learning patterns or dynamic characteristics to update his or her psychological model. Meanwhile, the students' responses to posed tasks are fed into, and diagnosed by, tutoring model. In turn, the student model is updated to reflect each student's current state. Based on the newly updated student model, the tutoring model adaptively selects appropriate an tutoring strategy, delivers appropriate learning content from the domain model, or gives adaptive feedback to the student. The function of each component will be described in the following subsections. The student model, the generic tutoring model and adaptive teaching strategies will be discussed in this chapter. Details about the domain model and the WAE are discussed in Chapter 4.

3.3 Building Blocks of the GWATS

3.3.1 Web-based Authoring Environment (WAE)

WAE is for instructors who do not have programming knowledge, as it provides a friendly environment for the easy creation of a web-based ATS applicable to various knowledge domains. WAE is the contributive component of GWATS that gives the GWATS the capacity to be used generically.

The authoring tools provide functions for educators to design domain-related knowledge and to create a student model for a specific domain. The generated domain model includes a hierarchical subject structure and related knowledge items stored in the database. The ATS creation process includes concept creation and editing, concept map compilation and generation, questions for creation/uploading and associations of the concept map with related learning materials. Of all the authoring procedures for an ATS, WAE is responsible for authoring the domain model and compiling the concept-network of the authored domain into the static student Bayesian network.

WAE might be the first system to employ Bayesian student modeling techniques in the authoring system, which uses a static student model to represent students' overall mastery states on each concept and a dynamic student model corresponding to students' responses to the assessment questions. A detailed description of WAE will be presented in Chapter 4.

3.3.2 User Interface

GWATS has two distinct types of users: authors and students. Hence, there are two types of interfaces: the authoring environment, which will be discussed in the next chapter, and the student interface through which students interact with the system. The user interface in GWATS is the major channel for conveying information and is a significant factor that affects users' learning performance, especially the speed and accuracy of locating particular information [109, 110]. The student interface outlines the functionality of the system and provides an interpretation of the learning environment from the students' point of view. A personalized learning environment should provide a flexible interface to create a comfortable environment accommodating students' individual preferences, keeping students informed about the adaptivity and options and providing them with control over the system [111].

The construction of a student interface is usually outside of the expertise of the potential ATS authors. GWATS provides an independent, general interface separated from the domain content for active learning. The student interface in GWATS provides an interactivity-rich environment with multiple types of learning methods tailoring to different learning styles.

The Welcome User page is how every user gains access, if the authentication is successful. The page provides an integrated environment for enhancing learning experiences. Through this interface, students can freely explore all the available tools and resources. The interface facilitates learning by providing several useful features. The author can access authoring and tutoring tools, as shown in Figure 3-2 . The student can only access the tutoring tools, as shown in Figure 3-3. Learning actions are tracked and recorded by the behavioral tracking module.

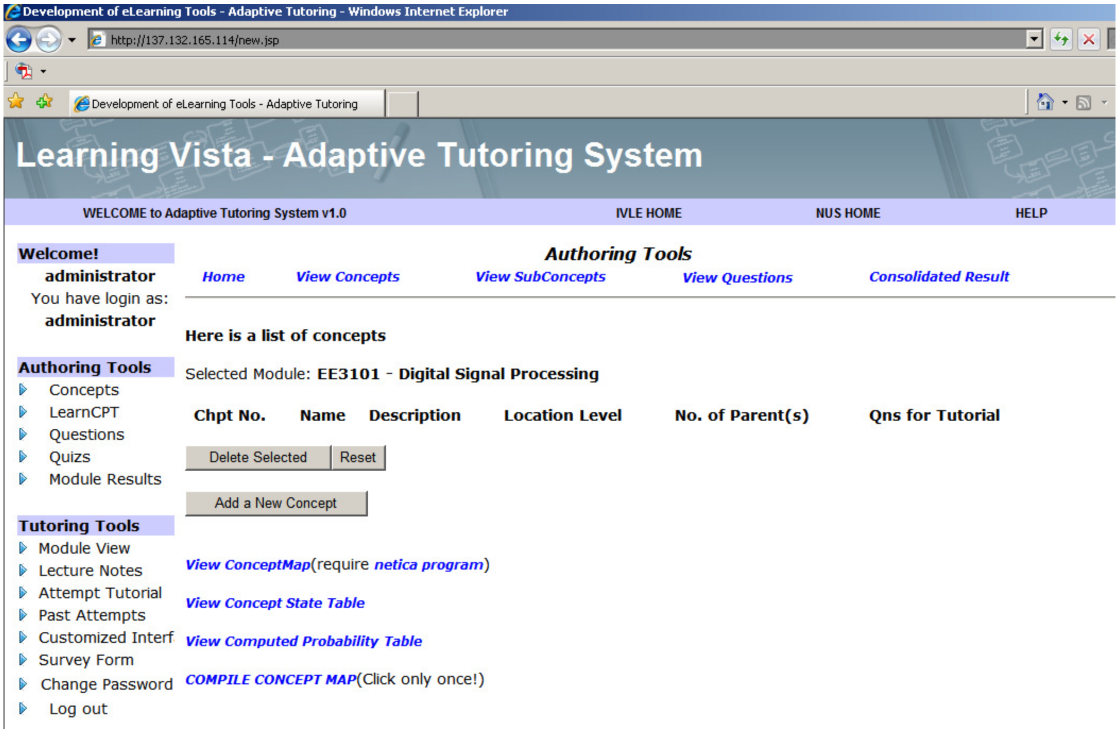


Figure 3-2: ATS author interface

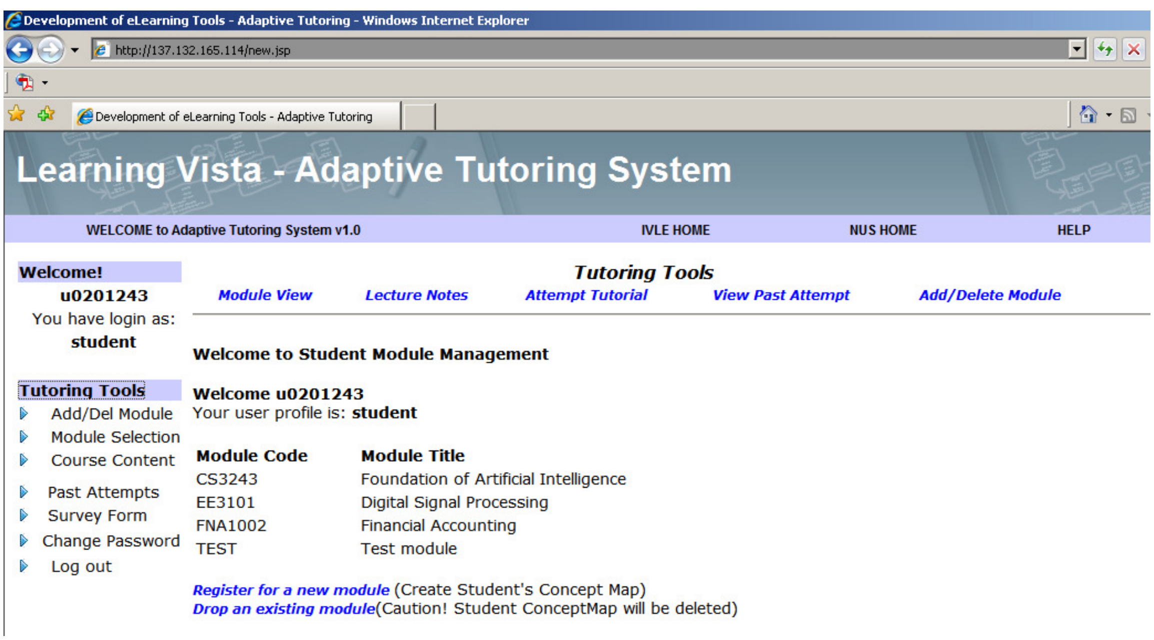


Figure 3-3: ATS student interface

3.3.3 Domain Model

Domain knowledge provides the context and domain information to be taught. The domain model governs the system's reasoning process. Accurate knowledge denotation of the domain model is required to accurately represent the chosen application domain. The domain model also supports important pedagogical actions, such as learning path organization, material delivery and feedback presentation. The ATS cannot function well without accurately representing knowledge about a teaching domain. Accurate definitions of the domain knowledge are one of the main contributions from the domain experts.

Within all of the representation techniques, the most widely used is the granularity hierarchical representation technique [112]. Granularity refers to the level of abstraction of the domain knowledge. Greer and his collogues [112] investigated a method for using different levels of granularity to conceptualize student knowledge. Their hierarchy consisted of various levels with decreasing degrees of abstraction from top to bottom. Therefore, granularity hierarchical structure models a domain from abstract to specific. There are two main relationships in the granularity hierarchical structure: prerequisite relationships providing knowledge unit ordering criteria and aggregation relationships breaking higher-level knowledge units into more fundamental sub-units.

Prerequisites help to guide student learning. For example, it may be desirable to begin to learn items of low difficulty. Depending on the student's performance with the selected concept, the ATS system can decide whether to move the prerequisite concept network up or down and can quickly focus on the current knowledge level of the student. Prerequisites also help to infer and deduce the achievement level of the student on concepts. The mastery of some concepts demonstrates the knowledge of its prerequisites. Conversely, from a non-mastered concept, the ATS may draw some conclusions about knowledge of a more difficult, follow-up concept. Based on the prerequisite relationships, the ATS can quickly and concisely deduce the mastery state of the student. We use a Bayesian network to represent the student knowledge model, and it is a natural way to compose a Bayesian network from a granularity hierarchical model and propagate evidence through the aggregation and prerequisite links.

In general, subject domain knowledge can be converted into a hierarchical representation, and four levels of granularities can be identified through which the student acquires concepts (Figure 3-4): subject (level 1), topics (level 2), concepts (level 3) and questions (level 4). Each subject is composed of many topics and is a structured hierarchy using aggregation relationships. Each topic can be decomposed into many concepts (a concept is deemed the basic knowledge unit in this research). Each concept has an associated set of questions that can be used to assess the student's knowledge level on the concept and all its preceding units. The knowledge units on the same level are organized as a network based on prerequisite relationships. Suppose there three nodes exist, $u, v, w \in N$, and there are connections between u and w and v and w. Then we have $(u, w) \in E$, and $(v, w) \in E$. This indicates that both concepts u and v should be known (i.e., learned) before learning concept w.

With this hierarchical representation architecture, to ease the evaluation process it is useful to partition the domain knowledge so the learning content at each level of granularity can be considered independently. The knowledge state of a higher level component can be inferred from the associated units on the following lower level. With a hierarchical structure, the system can infer mastery states of the higher unobservable knowledge unit from lower observable units. Therefore, all the knowledge unit/components in the granularity hierarchical structure can be evaluated directly or indirectly by the tutorial questions. Every knowledge unit is represented by a concept in the domain model.

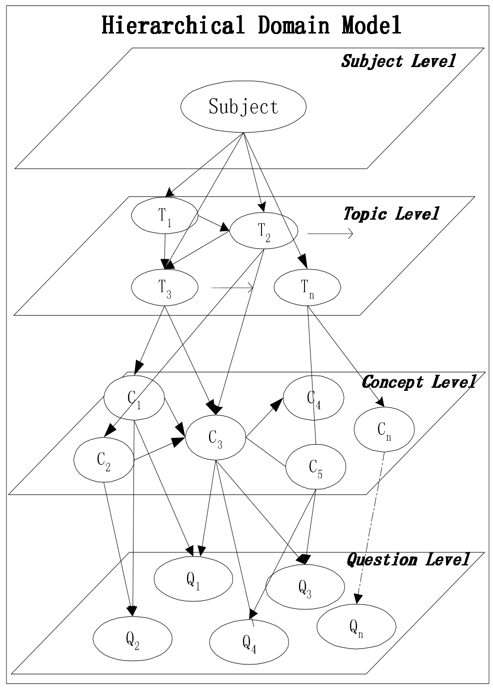


Figure 3-4: GWATS hierarchical domain structure

Only tags are used in the whole structure, while the corresponding learning materials are stored in the database. Therefore, the domain model in GWATS includes the hierarchical domain structure and the associated learning materials. It constitutes a multiple external representation of the concept, such as definition, concrete examples, exercises and questions for assessing the learning outcomes. GWATS provides an active learning environment by presenting multimedia learning materials-simulation, visualization, virtual laboratory, etc, such that the needs of students with different learning styles are met.

Like other intelligent educational systems built based on hierarchical architecture, GWATS in phase Ⅰ represents only the declarative knowledge, which is a sufficient prerequisite for procedural knowledge. In order to acquire procedural knowledge, it is necessary to understand the prepositional relationships between the concepts involved in procedural and declarative knowledge. Procedural knowledge is goal-oriented and mediates problem-solving behavior. It needs to be represented as a set of production rules that associate problem states and problem-solving goals with actions and consequent state changes. The price of such complex procedural knowledge modeling is a substantial decrease in its tractability. Hence, modeling the procedural knowledge in phase I of GWATS is not considered.

3.3.4 Behavior Tracking and Analysis Module

One of the contributions of GWATS is that it provides a behavior tracking module to track all data relevant to the student, not only consisting of the browsing behaviors, but also collecting all the historical data necessary for constructing the Bayesian student model. Consequently, GWATS provides students with two learning modes: student-controlled mode and the system-controlled mode. Under the student-controlled mode, the student is given a certain degree of control over the learning goals. After the learning goal is chosen, the system dynamically organizes and presents the learning materials based on the student model. During the learning process, the system monitors and records all the learning behaviors and activities, such as navigational actions, time taken and even mouse clicking actions to analyze, deduce and update the learning characteristics of the students. For a system-controlled mode, the system determines the appropriate learning goal based on the student model and uses the gradually improved learning characteristics to guide students.

We developed an event-tracking module to monitor and track all actions through the system interfaces. When an action occurs on the client side, the event-tracking module catches the actions of the student and sends the data to the server side for recording. All actions initiated by student, such as navigational histories, learning actions, time spent, idle time, etc., are recorded in the server database (Figure 3-5). The collected actions and behavior are fed to the behavior-analyzing component for information extraction and the results are used to update the individual student model.

Behavior analysis is based on the interaction collected by the behavioral tracking module. The purpose of this module is to interpret students' learning behavior, extract students' learning characteristics and update the student model to provide more tailored support to the students.

id	sURL	sLabel	sLinkID	start_time	user_id	module_id	concept_id
8846	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31375
8847	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31375
8848	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31375
8849	http://137.132.165	Example	example	Monday, March 5,	U0303985	EE4415	31375
8850	http://137.132.165	Example	example	Monday, March 5,	U0303985	EE4415	31375
8851	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31375
8852	http://137.132.165	Attempt	example	Monday, March 5,	U0303985	EE4415	31375
8855	http://137.132.165	Attempt	example	Monday, March 5,	U0303985	EE4415	31375
8861	http://137.132.165	Example	example	Monday, March 5,	U0303985	EE4415	31375
8862	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31375
8863	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31376
8864	http://137.132.165	Example	example	Monday, March 5,	U0303985	EE4415	31376
8868	http://137.132.165	Definition	link_0	Monday, March 5,	U0303985	EE4415	31307
8869	http://137.132.165	Example	example	Monday, March 5,	U0303985	EE4415	31307
9979	http://137.132.165	Attempt	example	Thursday, March 8	u0303985	EE4415	31375
9980	http://137.132.165	Start Tutor	link_3	Thursday, March 8	u0303985	EE4415	31375
9981	http://137.132.165	B	radio_5	Thursday, March 8	u0303985	EE4415	31375
9982	http://137.132.165	D	radio_10	Thursday, March 8	u0303985	EE4415	31375
9983	http://137.132.165	Hint	hint2	Thursday, March 8	u0303985	EE4415	31375
9984	http://137.132.165	Hint	hint2	Thursday, March 8	u0303985	EE4415	31375
9985	http://137.132.165	Hint	hint2	Thursday, March 8	u0303985	EE4415	31375
9986	http://137.132.165	Hint	hint2	Thursday, March 8	u0303985	EE4415	31375
9987	http://137.132.165	Hint	hint2	Thursday, March 8	u0303985	EE4415	31375
9988	http://137.132.165	Hint	hint1	Thursday, March 8	u0303985	EE4415	31375
9989	http://137.132.165	Hint	hint2	Thursday, March 8	u0303985	EE4415	31375
9990	http://137.132.165	Hints for Ta	button_3	Thursday, March 8	u0303985	EE4415	31375
9991	http://137.132.165	Submit	button_17	Thursday, March 8	u0303985	EE4415	31375
9992	http://137.132.165	Back to cor	link_3	Thursday, March 8	u0303985	EE4415	31375
9993	http://137.132.165	Attempt	example	Thursday, March 8	u0303985	EE4415	31375
9994	http://137.132.165	Start Tutor	link_3	Thursday, March 8	u0303985	EE4415	31375
9995	http://137.132.165	B	radio_5	Thursday, March 8	u0303985	EE4415	31375
9996	http://137.132.165	B	radio_8	Thursday, March 8	u0303985	EE4415	31375
9997	http://137.132.165	Submit	button_17	Thursday, March 8	u0303985	EE4415	31375
9998	http://137.132.165	E	radio_11	Thursday, March 8	u0303985	EE4415	31375
9999	http://137.132.165	Submit	button_17	Thursday, March 8	u0303985	EE4415	31375
10000	http://137.132.165	ASIC_Desig	link_0	Thursday, March 8	u0303985	EE4415	31375
10001	http://137.132.165	Attempt	example	Thursday, March 8	u0303985	EE4415	31375
10002	http://137.132.165	Start Tutor	link_3	Thursday, March 8	u0303985	EE4415	31375

Figure 3-5: Tracked learning behaviors

Two types of information can be obtained from the student: explicit and implicit. Explicit information can be directly obtained from the student's actions. Implicit information is hidden from the student and is therefore more difficult to obtain, requiring careful analysis of the behavior, which includes the student's emotional status, learning style, attitude, confidence and other psychological characteristics (Figure 3-6). For example, the analysis module can present the student with easier questions at the beginning and gradually increase the level of difficulty. The module can also provide encouragement or hints whenever it detects he or she faces difficulties in solving problems.

Category	Summary of data
Explicit	**Username, Page visited, Date & time of access, Time spent per action, exercises learned, questions attempted, hint request, actions before and after errors, Link clicked, Button clicked,etc**

Category	Summary of data
Implicit	**Confidence, effort, attitude, independence, emotion status, learning patterns, preferred learning styles, etc.**

Figure 3-6: Behavior analysis

Statistics, machine learning, data mining and other technologies can be used for the analysis to gain insights into students' psychological states, strategies applied and learning patterns. The ultimate goal of our approach is to assist students in improving their learning behavior.

3.3.5 Student Model

The student model contains detailed information. The suitability and accuracy of the student model influences the effectiveness and the efficacy of the adaptive tutoring [113]. However, the significance of the student model is how the tutoring model makes use of this detailed information to provide adaptive tutoring. The model is the driving force that enables the GWATS to provide personalized tutoring.

The traditional ITS student model represents the student's knowledge of the skills, concepts or learning objectives in the domain model. The GWATS model takes in account many discriminative characteristics relevant to learning, ranging from prior knowledge of the domain and knowledge of the units in the domain model, to learning preferences, styles and motivational states, etc. The GWATS model includes two main components with two distinctive types of information: 1) The domain-related knowledge component reflecting the system's interpretation of the student's mastery and 2) the domain-independent psychological component describing the student's domain-independent characteristics (confidence, emotion, learning style, etc.). A student's knowledge level changes periodically and can be determined via his or her interactions with the system as he or she tries to handle the educational tasks. The structure of the knowledge model tends to be stable, but the knowledge states change over time as the student accumulates knowledge. The psychological characteristics are used as a guideline to keep the student in an optimal learning state. Identifying and reacting to students' psychological states in an adaptive tutoring system have gained extensive attention in recent years. A student's psychological characteristics might also change in time. In GWATS, the psychological characteristics are extracted by the behavioral analysis model based on tracked behaviors.

The overlay model technique considers a student's knowledge as a subset of the expert's knowledge, and the student model is a subset of the expert model [114,115]. GWATS employs a granularity hierarchical architecture with four levels to structure domain knowledge and divides this knowledge into generic items. Since the student knowledge model represents an estimation of the knowledge level for each knowledge unit, it is obvious that the student knowledge model has the same typology as the domain model. The tutoring model can infer another knowledge unit, whichever level it is on directly and indirectly from the knowledge states of concepts, thanks to the prerequisite and aggregation relationships. GWATS employs a knowledge model that is similar to the domain topology and records a student's knowledge level of the concepts. This is a very natural way to use a Bayesian network to represent a student model with nodes corresponding to concepts and links reflecting prerequisite relationships among concepts.

A number of adaptive education systems have utilized a Bayesian network in student modeling [116-118]. These mainly use the Bayesian student model to analyze a student's knowledge states on the question level during the decision-making process. The Bayesian student model in GWATS consists of two parts: static and dynamic. The static part contains a student's overall mastery states of each concept in the module, corresponding to the overall concepts network of the whole module. This information is maintained and updated during the learning process. The dynamic part is automatically generated when the student chooses to attempt a target concept, including the concept, the assessment questions and all related concepts. Questions are selected by teaching strategies to assess the student's mastery of the target concept. Mastery of the higher-level objects are deduced from the concept mastery values. How the static and dynamic student parts serve as the basis for providing personalized tutoring is presented in the next subsection.

Let $Gdm = (V, E)$ serve as a prerequisite network to model the domain, where $V = (C_1, C_2, ..., C_{n\text{-}1}, C_n)$ is the ordered set of concepts in this model and E is a set of edges. We can describe an individual's knowledge about the domain at a particular timestamp t as an ordered set $SM_t = (S_1, S_2, ..., S_{n\text{-}1}, S_n)$, where each element S_i represents the learning state regarding to its corresponding concept in V. That is, $S_k \in$ {Mastery, Partial-Mastery, Non-Mastery}, $k = 1, 2, ..., n$. Note that $|SM_t| = |V|$, thus the student's knowledge status of the domain is exactly overlaid on the domain model. Although there are only three possible learning states for each concept here, there are numerous approaches to determine the learning states of certain concepts according to the tutoring model.

Figure 3-7 shows a dynamic Bayesian network created for a tutorial. The targeted concept of the tutorial is concept *C1*. Figure 3-8 shows the Bayesian network after attempting the tutorial (i.e., after adding evidence to the question nodes).

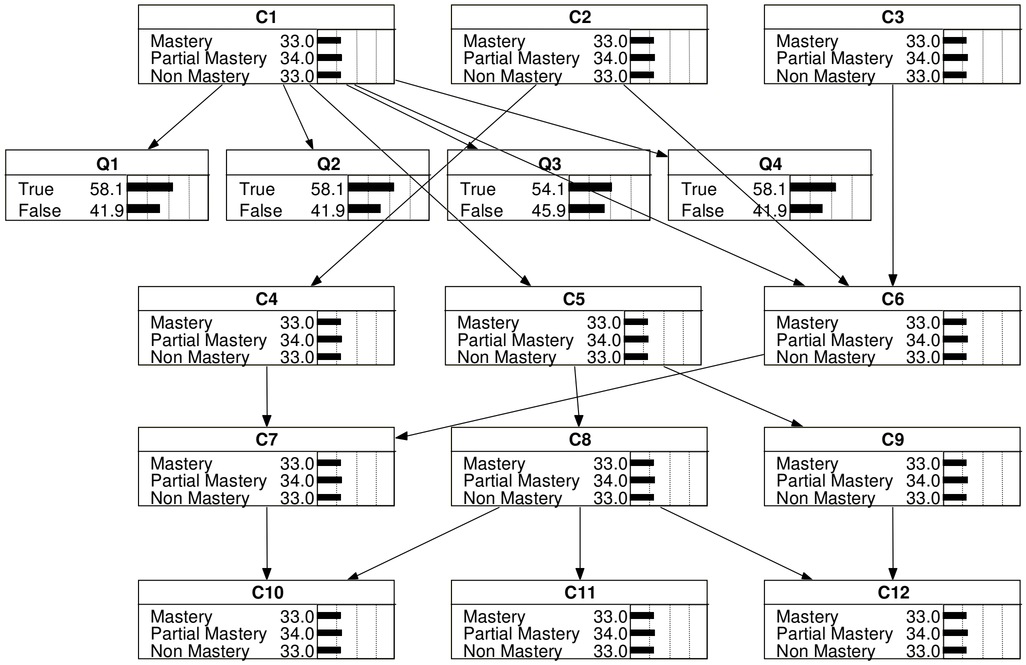


Figure 3-7: New Bayesian network created for a tutorial before adding evidence

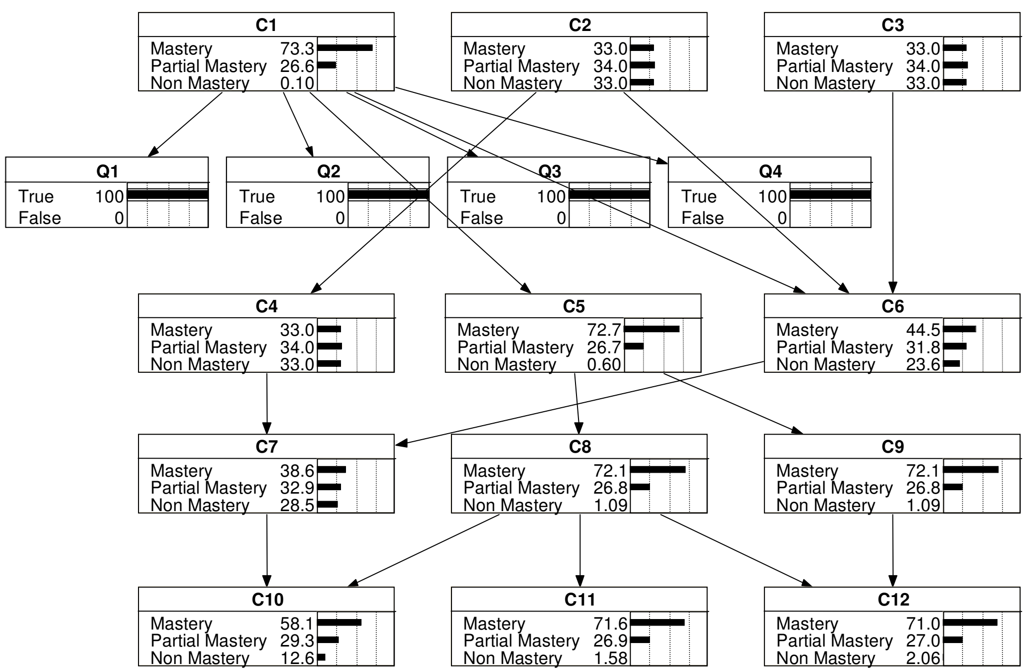


Figure 3-8: New Bayesian network created for a tutorial after adding evidence

When the network in Figure 3-7 is compared to the network in Figure 3-8 it can be seen that although all the questions of the tutorial are only related to concept *C1*, the results of the tutorial has affect on the states of other concepts related to *C1*. However, only the targeted concept is taken as mastered by the student. The changes in the other concept states will be used for the concept selection and question selection algorithms as discussed in later sections.

Figure 3-9 shows a Bayesian network created for a tutorial targeting concept C12. Questions in this tutorial belong to many concepts and the states of those concepts will change accordingly. When there are questions related to more than one concept, the states of all the related concepts will be updated unlike in the example given in Figure 3-9 The questions for each tutorial are selected using one of the three question selection algorithms: random, information gain method and conditional probability method. The next chapter describes these algorithms in more detail.

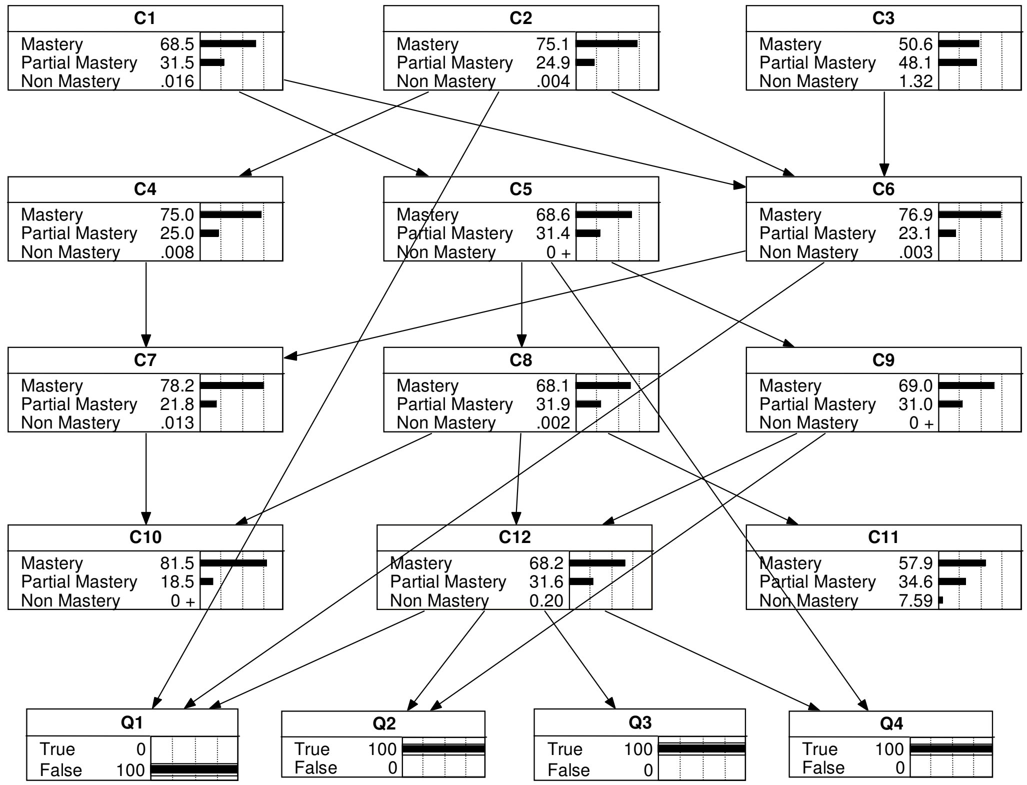


Figure 3-9: A Bayesian network of a tutorial with questions belonging to more than one concept

3.4 The Use of Generic Tutoring Model

The tutoring model in GWATS depends on the general structure of application domains and the uniform framework of the student model. The separation of the tutoring strategies from the domain description, which is similar to the separation of the knowledge representation technique from the learning material, promotes the reuse of strategies over various domains as long as they can be organized in the same granularity hierarchical structure.

The tutoring model actually is a back-end engine that uses the student model as a guideline for decision making and providing feedback in the tutoring process based on the exploited teaching strategies. Teaching strategies are the way the tutoring model takes control of the interaction with students. The responsibility of the tutoring component is to employ the most appropriate teaching strategy to the specific student in a given context. A good tutoring system should provide multiple teaching strategies to accommodate a variety of students with different characteristics and various needs. The integration of ITS technology with AHS allows GWATS to provide personalization and adaptation, based on the ITS tutoring techniques and the AHS adaptation support technologies in a number of ways: concept selection, diagnosis of knowledge level, question selection, adaptive feedback and adaptive presentation.

The tutoring model makes these decisions based on the static and the dynamic Bayesian student models. A static Bayesian student model maintains the student's real-time mastery of each concept and is used as benchmark to organize an individual learning path to achieve the learning objective chosen by the student. A dynamic Bayesian student model contains the posed questions, the concept to be assessed and all the related concepts. It is used to identify the student's misconception after each tutorial and as evidence used to update the static student model. In addition, the tutoring model keeps a temporary Bayesian network with all the concepts and the target concept-related questions to select questions adapted to the individual's states and needs. The details of decision making will be described in the following subsections.

3.4.1 Learning Path Organization

This function works based on the prerequisite relationships between learning objectives. When a student selects a learning objective, the tutoring model will check if the student lacks any prerequisite knowledge on the unit and will present a test for the lacking prerequisites, one by one, based on the difficulty of the concepts. If the student passes the supplementary tests, the model indicates that the student has sufficient information on the prerequisite concept. Only if all the prerequisites are mastered can the student can get access to the chosen objective.

Two types of learning objectives can be set up: concept and topic. It is easier if the objective is a concept, and we will use the breadth-first-traversal method starting from the destination until arriving at concepts that the student has mastered. The breadth-first method selects the concepts at the same level in the course hierarchy concept (topic) as the next recommended concept. This method makes sure that the student meets all the prerequisites before moving on to the more complicated concepts. During the traversal, all the passing un-mastered concepts are pushed in the stack. After the traversal is over, the learning path is generated by popping the concepts in to the stack, one by one. The student can achieve the learning objective by learning the start concept through to the last one.

If a topic is set as a learning objective, we should use the aggregation relationship between the topic level and the concepts to get all concepts related to the chosen topic. The learning path organization algorithm for a topic as a learning objective is as shown in Figure 3-10.

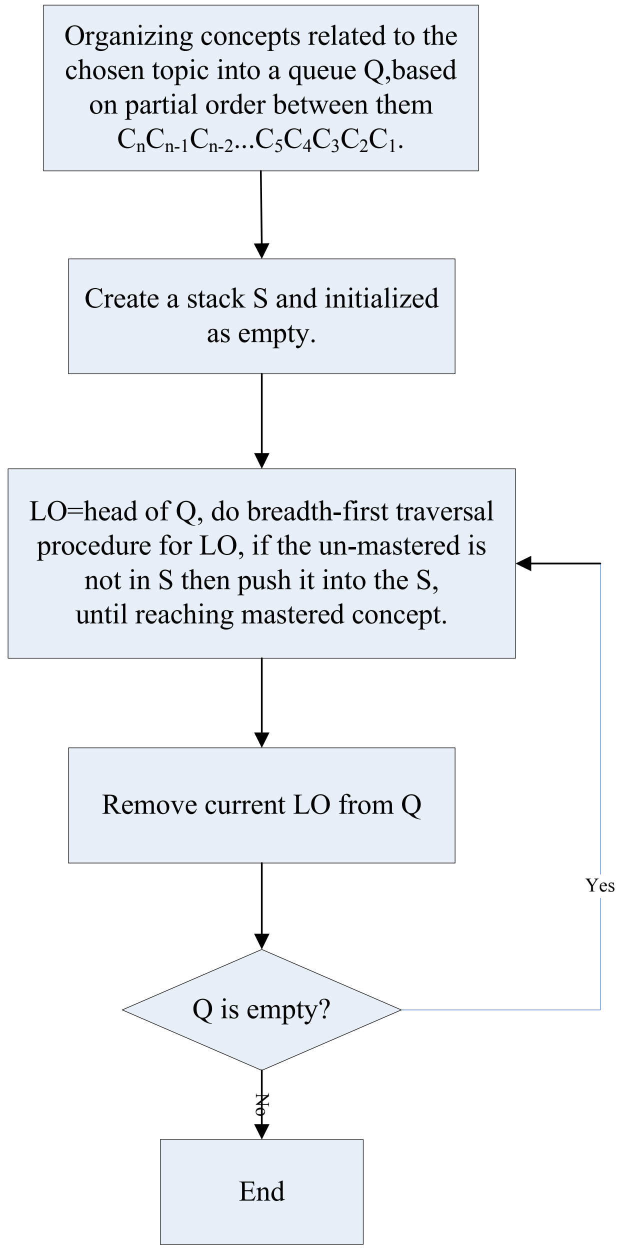


Figure 3-10: Learning path organization algorithm

After this process is over, a learning path is generated by popping the concept in to the stack, one by one. Students can achieve the chosen learning objective by learning through the generated path.

3.4.2 Adaptive Delivery

Adaptive delivery means that tutoring model that presents the most appropriate tutoring to students. The tutoring model ensures that a student is allowed to access a concept only if the parent concepts have been mastered or partially mastered. The most appropriate concept to be learned next depends on the student's knowledge states for all concepts based on the static student knowledge model. The concepts not learned, whose parents were learned and so the student is in a partial-mastery or mastery state, with the minimum location level, are chosen as the next targets.

Students in a large class may have different learning preferences and may learn more efficiently than others due to their individual learning styles. According to the theory of learning styles [126-127], the effectiveness of teaching and learning is mainly influenced by educational experiences geared toward students' styles of learning. It is crucial to identify an individual's learning style before providing personalized learning adapted to his or her style. But a learning style is difficult to recognize. The style-matching strategy is based on the assumption that a style has temporal stability and an individual's style will remain relatively constant for a period of time, which has not been proven by research to date [71]. GWATS takes the complexity of learning styles and the difficulty of identification into account and provides as many types of learning materials and methods as possible, as well as the freedom for individuals to choose, to cater to students with different learning styles, including definitions, examples, exercises, simulations, visualizations, virtual laboratories, etc.

Meanwhile, GWATS is designed to present an interactive and hands-on learning environment to facilitate active learning. Simulations provide a controlled environment in which students solve problems with the application of knowledge gained to manipulate input variables and observe the consequences, to formulate and test hypotheses and to interpret principles or properties displayed by models. Visualizations give students a clear picture of the underlying concept for a mathematic description and stimulate their curiosity and thinking ability. A virtual laboratory motivates students to conduct experiments using a web browser and provides simulations of complex processes similar to those in a real laboratory. These multimedia learning methods not only enable students learn in their preferred ways and cater to different learning styles, but also encourage them to learn through the Kolb's learning cycle and engage in active learning.

3.4.3 Question Selection

After the learning path is organized for each individual, the student needs to learn through the concepts, one by one, along the path to achieve the objective. Questions are the best way to assess students' knowledge state and diagnose their mastery of each concept. Each question in the GWATS is associated with one or more concepts. This association tells how much this question provides information and evidence about the student on the assessed concept. The tutoring model selects the most adequate questions to obtain an estimation of student knowledge about the target concept. Question selection is a very challenging task, because students at higher levels can become bored if they are presented with questions that are too easy for them, and those at lower levels can be discouraged if presented with questions that are too difficult.

Most of the research uses the Item Response Theory (IRT) as the underlying model to select questions. It is assumed that the knowledge level of a student is measured with a single variable θ , which is called the trait. It expresses a logistic relationship between the student's mastery of a concept and the probability of a correct response. The IRT method is sufficient for evaluation purposes, but might be problematic if more information about a student is required [128]. In order to know which concepts are difficult for a student in addition to the overall mastery states of a concept, we used the Bayesian network as a theoretic basis for student modeling and decision theory to select the next questions.

To assess the mastery states of the students quickly and accurately while not frustrating or boring them, various considerations were taken into account. The most important factors are who is to be assessed (specific information about a student), what is to be assessed (appropriate learning content for a student) and how to assess (criteria for selecting content).

The tutoring model selects the next group of questions depending on the estimated knowledge levels of a student on the target concept and related concepts. The chosen questions will adequately discriminate the different mastery states and determine the exact knowledge level of the student based on his or her answers.

First, questions should be selected according to the current estimation of the student's knowledge. In GWATS, this criterion means that the selected questions must cover all the prerequisite concepts in mastery or partial mastery states, except for the target concept. Second, the questions should be selected based on the performance previously shown by the student. The ideal questions should be informative enough to distinguish mastery from non-mastery of the target concept.

Procedure for selecting questions is as follows:

1) Filtering questions with related concepts; only target concept-related questions are selected.

2) Only un-attempted questions are candidates.
3) Parents' states must be partial mastery or mastery
4) Using question selection criterion to select questions.
5) Number of questions required to assess the target concept and the level of difficulty are taken into consideration.

The filtering step reduces the number of candidate questions before involving adaptive question selection methods, therefore the time and computation cost of queries in a Bayesian network decrease remarkably.

There are numerous measures for selecting the most appropriate questions, such as minimum expected cost, information gain and maximum discrimination. We used information gain and the maximum discrimination methods based on a temporary Bayesian network. The details about this network and the implementation details about the two question selection methods are described below. The efficiency and effectiveness of these two methods are evaluated in Chapter 5.

Note that in this thesis, questions refer to closed questions used to assess knowledge units. Closed questions can be useful in eliciting a quick response. In contrast, open questions might be more effective in evaluating high-level objectives. However, open questions are very difficult to handle. The evaluation of open questions is extremely time-consuming and difficult even for actual teachers. Therefore, we excluded the open question discussion from this thesis. Besides, ATS is not a replacement for human tutors, but provides a supplement to regular tuition to reinforce the concepts taught in a large classroom. We will leave the high-level open questions for human tutors to deal with.

3.4.3.1 Temporary Bayesian Network Creation

Each to-be-assessed concept has one corresponding temporary Bayesian network used as the basis for choosing the most appropriate questions. The temporary network includes these to-be-assessed concepts, all its parents and all related concepts. All the probabilities of the concepts can be copied from the static student model. The conditional probabilities of the questions' related concepts are retrieved from the database. The temporary network is automatically created each time a student chooses to attempt a quiz about the target concept and will be discarded after the questions are selected.

3.4.3.2 Selecting Question Based on Information Gain Theory

The principle of selecting questions based on information gain involves choosing the questions that will maximize the expected reduction of entropy of the test using the measure of information from information theory and Shannon entropy [129]. Entropy is essentially a measure of the amount of uncertainty in a system of stochastic events [130].

Note: Each concept can be in any of the three knowledge states: mastery ($C_k=M$), partial mastery ($C_k=PM$) and non-mastery ($C_k=NM$); each question can be answered correctly ($Q_i=T$) or incorrectly ($Q_i=F$). The following steps explain the process needed to select the best questions based on a student's past performance and the information gain method.

1) Find the prior probabilities.

The prior probabilities that a student having a mastery state of k – $P(C_k)$ on a concept and the probability that a question answered correctly or incorrectly given a mastery state of k $- P(Q_i \mid C_k)$ need to be determined.

Initially, all students share the same prior probabilities of $P(C_k)$, which have a normal distribution. As the tutoring interaction progresses, $P(C_k)$ will be automatically and dynamically updated based on the new evidence—the student's corresponding updated probabilities of the assessed concepts at the end of each tutorial. $P(Q_i \mid C_k)$ can be calculated for each question by setting the mastery states of the concept to all of the three possible states.

2) Get the value of the normalizing constant c

$$c = \frac{1}{\sum_{k=1}^{3} P(Q_i \mid C_k) P(C_k)}. \tag{3-1}$$

c can be calculated using this formula based on the prior probabilities calculated in the previous step. The normalized constant is needed to make sure that the sum of all the posterior probabilities is equal to 1. For example, c_T for correctly answering a question can be calculated as follows:

$$c_T = \frac{1}{P(Q_T \mid C_M) P(C_M) + P(Q_T \mid C_{PM}) P(C_{PM}) + P(Q_T \mid C_{NM}) P(C_{NM})}. \tag{3-2}$$

3) Calculate the posterior probabilities.

The posterior probabilities of the estimation of mastery states of the objective concept in k given his or her response i can be found using the Bayes' Theorem as shown below:

$$P(C_k \mid Q_i) = CP(Q_i \mid C_k)P(C_k). \tag{3-3}$$

The conditional entropy for correctly and incorrectly answering a question is calculated by using the formula given below:

$$H(S) = \sum_{k=0}^{1} -P_k \log_2 P_k. \tag{3-4}$$

Therefore, the conditional entropy for correctly answering a question is calculated as follows:

$$\begin{aligned} H(S_i \mid Q_T) = [&-P(C_M \mid Q_T)\log_2 P(C_M \mid Q_T) - P(C_{PM} \mid Q_T)\log_2 P(C_{PM} \mid Q_T) \\ &-P(C_{NM} \mid Q_T)\log_2 P(C_{NM} \mid Q_T)]. \end{aligned} \tag{3-5}$$

4) Find $P(Q_i)$ using the following formulas:

$$\begin{aligned} P(Q_T) = [&P(Q_T \mid C_M)P(C_M) + P(Q_T \mid C_{PM})P(C_{PM}) \\ &+P(Q_T \mid C_{NM})P(C_{NM})], \end{aligned} \tag{3-6}$$

$$\begin{aligned} P(Q_F) = [&P(Q_F \mid C_M)P(C_M) + P(Q_F \mid C_{PM})P(C_{PM}) \\ &+P(Q_F \mid C_{NM})P(C_{NM})]. \end{aligned} \tag{3-7}$$

5) Weigh the conditional entropies using the formula given below:

$$H(S_i) = P(Q_T)H(S_i \mid Q_T) + P(Q_F)H(S_i \mid Q_F). \tag{3-8}$$

Calculate the entropies from Steps 1 to 5 for every question, then select those with the greatest difference between $H(S_0)$, the conditional entropy of correctly answering a question, and $H(S_1)$, the conditional entropy of incorrectly answering a question. The entropies of each question are calculated in Equation 3.8. Until now, all calculations could be performed by doing queries of the temporary Bayesian network. Equation 3.3 is usually used to determine the student's mastery states of the related concept C based on his or her responses to the posed questions.

3.4.3.3 Selecting Question Based on the Conditional Probability

Each question in GWATS may be related to more than one concept. The purpose of the students attempting a tutorial after they learn the concept is to measure whether they mastered the target concept. Answers to questions are evidence of a student's knowledge of the related concepts. This evidence is used to determine the following learning objective and path. The objective of selecting questions based on the conditional probability method is to select a set of questions best-suited for testing. The suitability of a question to the target concept is defined as:

$$P(C_M \mid Q_i) = P(C = M \mid Q_i = 1), \tag{3-9}$$

which means the probability of C is mastered, given Q_i is correct and

$$P(C = M \mid Q_i = T) = \frac{P(Q_i = T \mid C = M)}{P(Q_i = T)}, \tag{3-10}$$

of which

$$\begin{aligned} P(Q_i = T) &= P(Q_i = T \mid C_M)P(C_M) \\ &+ P(Q_i = T \mid C_{PM})P(C_{PM}) + P(Q_i = T \mid C_{NM})P(C_{NM}). \end{aligned} \tag{3-11}$$

All of the calculations in these equations can be performed by querying the temporary Bayesian network. Compared to the information gain method, the conditional probability method requires fewer computations. All of the values required can be computed at the start of the tutorial and therefore no Bayesian network has to be created while the student is attempting a tutorial. The evaluation of these two question selection methods using simulated students is presented in Chapter 5. Both methods show predominance over the random question selection method. We chose the conditional probability method for this thesis because it requires fewer computations.

3.4.4 Estimation of Student Knowledge Status

Due to the hierarchical structure of the student model, assessments can be performed at any level of granularity within this hierarchy. However, since there are prerequisite relationships among concepts, the information of the student mastering one concept may have an impact on several other concepts. Knowledge can be propagated throughout the concept's Bayesian network based on the concept's granularity hierarchy. Basically, three possible states exist for each learning object: non-mastery, partial mastery and mastery. For each concept, the mastery state is updated by the student's answers to posed questions. The estimation of the student's mastery state is formed using the priors and observations according to Bayes' theorem,

$$P(C_K \mid Q_i) = cP(Q_i \mid C_K)P(C_K), \tag{3-12}$$

where subscription K=M or PM or NM. M, PM and NM represent mastery, partial mastery and non-mastery, respectively. C_K represents the mastery state of concept C. $P(C_K)$ is the priori probability of concept C, which can be copied from the static student model. $P(Q_i \mid C_K)$ is the conditional probability of the student correctly answering question Q_i given the mastery state of C_K, which can be copied from the dynamic student model. For each concept, there are three probabilities, one for each mastery state. The rule for classifying an examinee based on these three probabilities is to select the category with the maximum a posterior probability. In GWATS, the tutoring model calculates the posterior probabilities for each mastery state given students' responses to the questions $P(C_M \mid Q_i)$, $P(C_{PM} \mid Q_i)$ and $P(C_{NM} \mid Q_i)$ $\mathrm{P}(NM|Z)$, then it selects the category with the maximum a posteriori probability as the most likely category.

The student's responses to posed questions are not only evidence about his or her knowledge of the directly evaluated concepts, but also on those that have relationships with directly evaluated concepts. With the evidence entered into the dynamic student model, the concepts propagate within the dynamic Bayesian network. The probabilities of each concept within it are updated. The updated probabilities of all concepts within the dynamic student model are considered as evidence for updating the static one.

3.4.5 Adaptive Presentation

GWATS employs the adaptive link annotation to determine the suitability of the link destinations. When a page with concepts in the domain is generated, links are displayed depending on the suitability of the link destination. The link is shown in blue when the student has mastered all the prerequisite parent concepts and is eligible to learn the destination concept, and the link is in black for other way around as shown in Figure 3-11. The accessibility of the concept will be dynamically updated by the tutoring model based on the student's mastery states of the concepts. The currently non-accessible concepts become accessible when all of its parents are learned and mastered or at least partially mastered by the student.

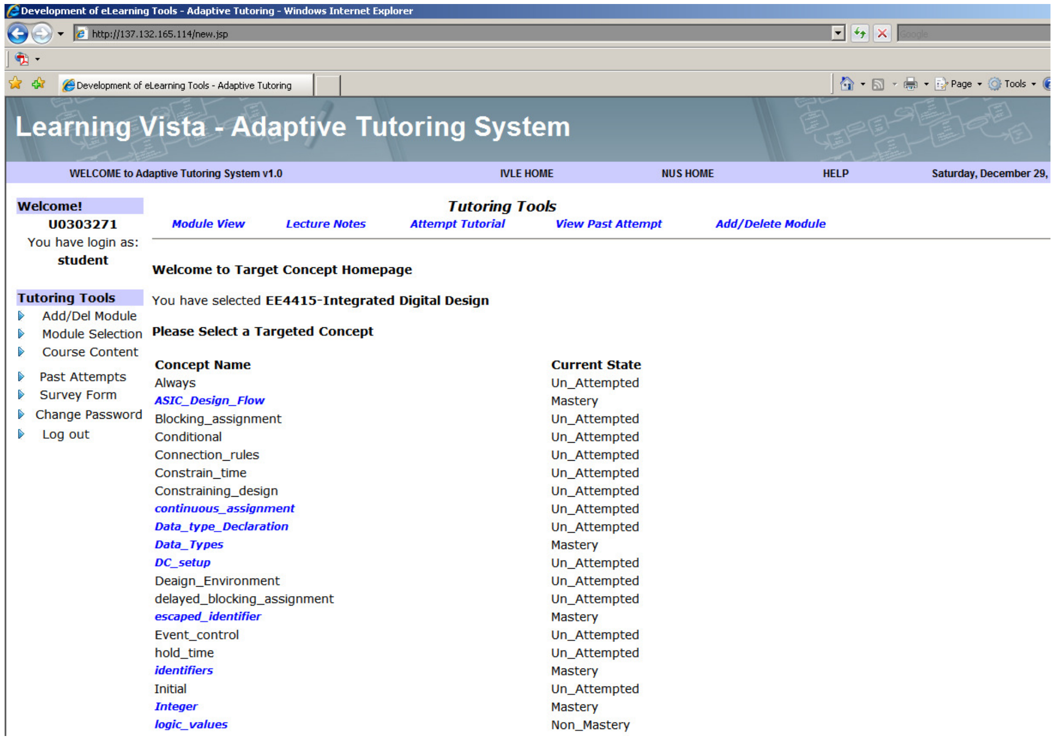
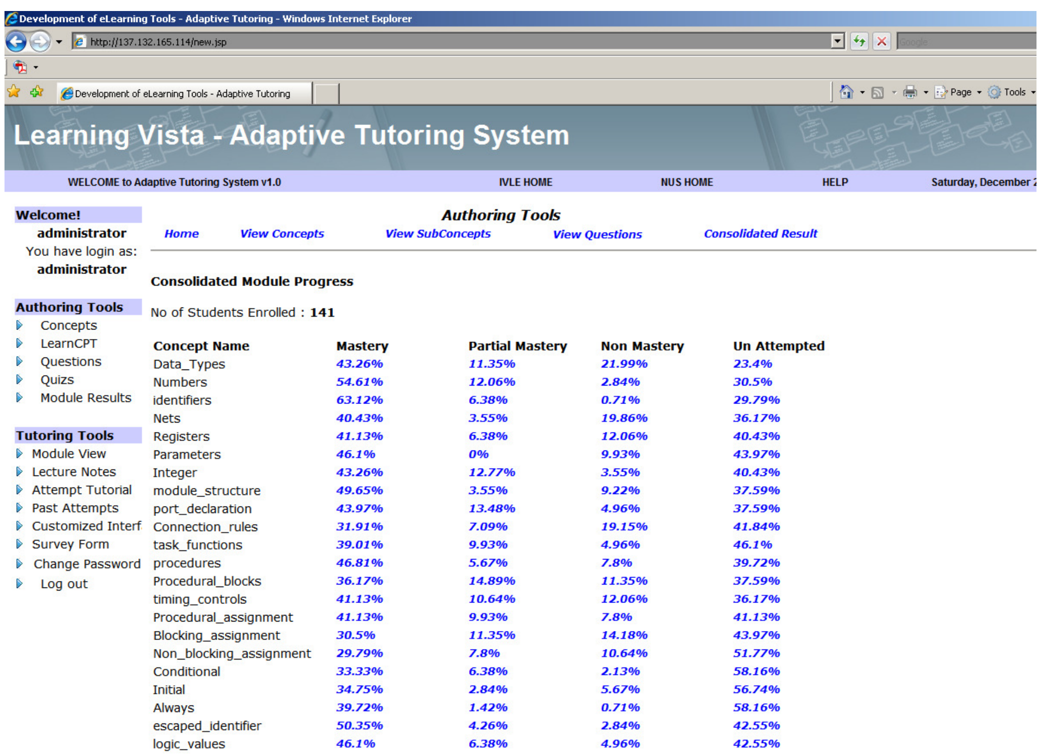


Figure 3-11: Concept selection interface

Once a student selects an accessible concept to study, the tutoring model will automatically select corresponding educational materials and load them into the framed learning environment and present them as appropriate for the student.

3.4.6 Adaptive Feedback

Feedback is the system's reaction to the student's learning behavior, which is one of the key ingredients offered by an e-learning system. The important role of feedback has long been recognized by educational researchers [131]. In most ITSs, feedback is an immediate reaction to the actual problem solving and is supposed to create a timely feedback loop between the system and the students. Basically, the two types of feedback are to the educators and to the students. The main functions of feedback are to help educators to know what difficulties students may face in their learning process, and to highlight strengths and weaknesses for students' further study and performance improvement. GWATS provides both types of feedback mechanisms. The consolidated feedback to lecturers presents the statistics for the mastery states of each concept for all students (Figure 3-12), students at each mastery state (Figure 3-13) and the learning progress for each student (Figure 3-14). Feedback to lecturers provides the general learning states of the classes and allows lecturers to adjust their teaching focus and pace for the benefit of the class. It also shows the performance of each student, which enables the lecturer to provide corresponding remedial tutoring to better suit a student's state and needs or sends suggestions or appropriate learning materials to the student.



Concept Name	Mastery	Partial Mastery	Non Mastery	Un Attempted
Data_Types	43.26%	11.35%	21.99%	23.4%
Numbers	54.61%	12.06%	2.84%	30.5%
identifiers	63.12%	6.38%	0.71%	29.79%
Nets	40.43%	3.55%	19.86%	36.17%
Registers	41.13%	6.38%	12.06%	40.43%
Parameters	46.1%	0%	9.93%	43.97%
Integer	43.26%	12.77%	3.55%	40.43%
module_structure	49.65%	3.55%	9.22%	37.59%
port_declaration	43.97%	13.48%	4.96%	37.59%
Connection_rules	31.91%	7.09%	19.15%	41.84%
task_functions	39.01%	9.93%	4.96%	46.1%
procedures	46.81%	5.67%	7.8%	39.72%
Procedural_blocks	36.17%	14.89%	11.35%	37.59%
timing_controls	41.13%	10.64%	12.06%	36.17%
Procedural_assignment	41.13%	9.93%	7.8%	41.13%
Blocking_assignment	30.5%	11.35%	14.18%	43.97%
Non_blocking_assignment	29.79%	7.8%	10.64%	51.77%
Conditional	33.33%	6.38%	2.13%	58.16%
Initial	34.75%	2.84%	5.67%	56.74%
Always	39.72%	1.42%	0.71%	58.16%
escaped_identifier	50.35%	4.26%	2.84%	42.55%
logic_values	46.1%	6.38%	4.96%	42.55%

Figure 3-12: Consolidated results

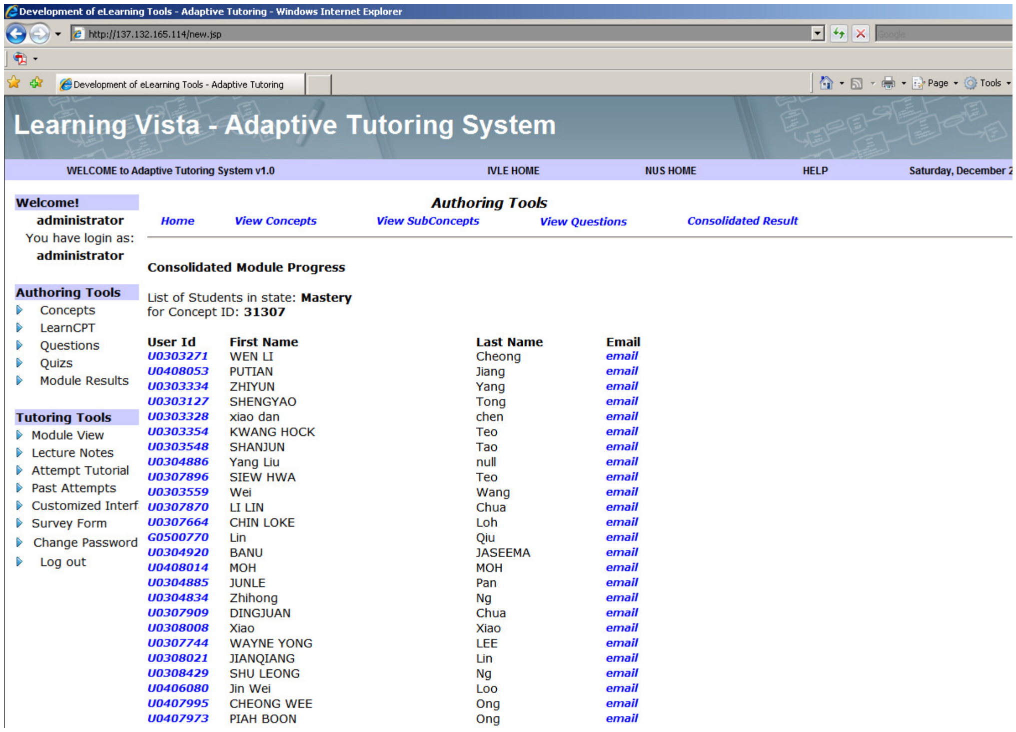


Figure 3-13: Students list in each mastery state

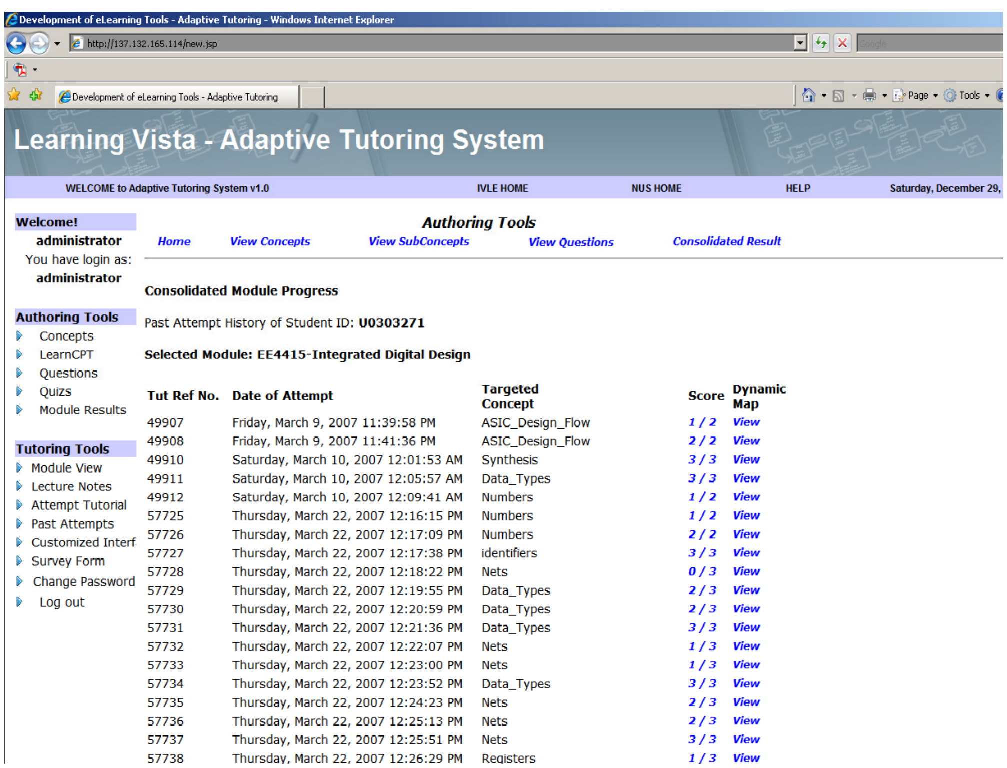


Figure 3-14: Students attempting history

Feedback in ITSs is usually designed for direct responses to students' actual problem-solving actions and to help them accomplish solutions in the cognitive tutor. For example, ANDES [100] provides procedural and error help by comparing a student's problem-solving steps with paths in a completely represented problem solving search space. This feedback provides information on the students' correctness and necessary hints or help to encourage them to work out as much of the answer on their own as they can, and to ensure that the students successfully complete each question. However, the authoring of this specific feedback and correct or erroneous steps is a very laborious task, which requires considerable effort to explicitly define what can go wrong and what the reasons are for each erroneous action. To avoid such a situation, we identified students' errors or misconceptions by linking the students' solutions as evidence to the corresponding dynamic Bayesian network and using the diagnosis capability of the Bayesian network to estimate the most likely cause of the errors.

Since one question may be related to more than one concept, there may be some confusion resulting in student errors. After each student's response, the dynamic Bayesian network enters the correctness/incorrectness as evidence and updates the network. The mastery probabilities of the target concept and other associated concepts of the student from these observable evidence are inferred and updated in the dynamic student model. Then the concept that shows a decrease in mastery probability and for which posterior mastery probability is close to zero is diagnosed as the weak concept and remedial action might be needed. In GWATS, feedback to students allows them to browse their own learning progress, to check their tutorials attempted and see the analysis of each tutorial about the possible misconceptions identified by the tutoring model (Figure 3-15).

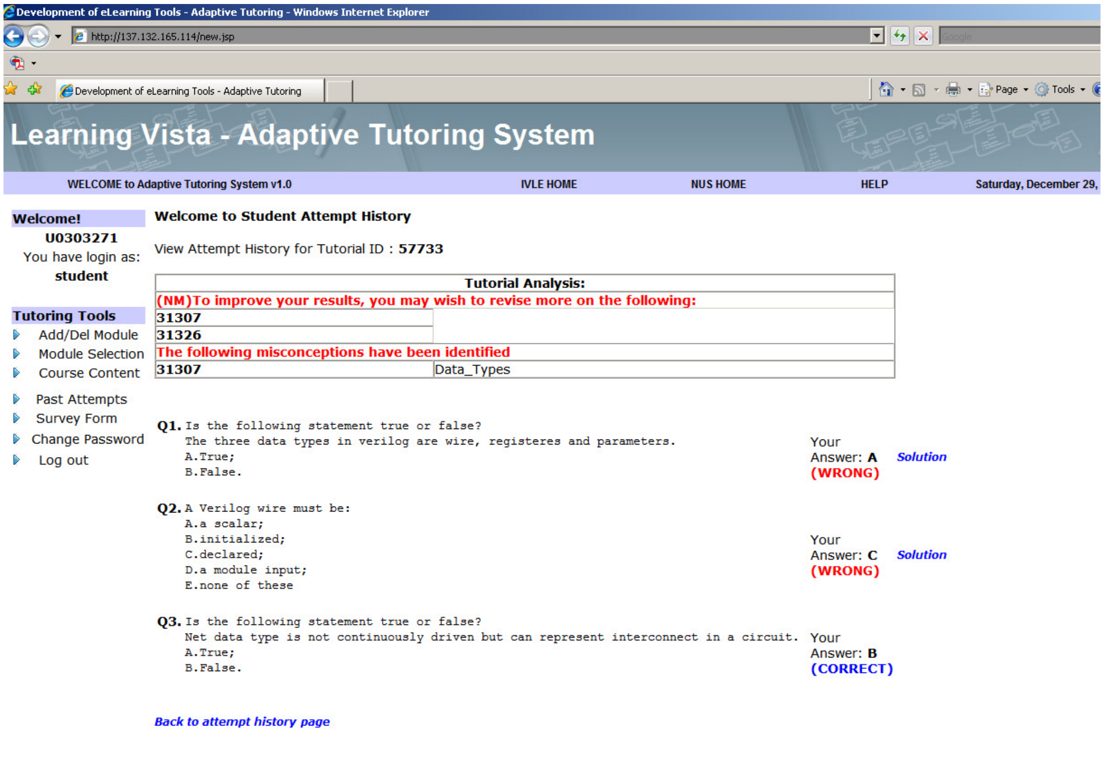


Figure 3-15: Tutorial feedback to the student

3.5 Conclusion

This chapter presents the design considerations of GWATS, discusses the benefits of integrating a WAE into its architecture, describes its architecture, elaborates on its basic components and highlights its back-end generic tutoring model and tutoring strategies. To create an ATS that is reusable in different domains, there is a trade-off between the generality of the authoring tool and the power of the constructed system. At present, all of the constructed ATSs authored with GWATS share the general tutoring strategies applicable to a wide range of domains. The generic tutoring model separates the tutoring strategies from a specific domain description and depends on the general domain structure and uniform student model framework. This framework provides a good approximation of the student's knowledge in the domain model and the learning traits in the psychological model and is essential in that it provides guidelines for personalized tutoring decisions made by the tutoring model. The genetic tutoring model provides openness to certain extent. It takes control and puts restrictions on resources that the students are not ready to attempt. It allows students to explore the contents freely while guiding them to accomplish learning tasks using an appropriate learning style. It will take control and illuminate them when they have difficulties. The tutoring model coordinates students' learning by reinforcing their strengths and helping them to overcome their weaknesses. It achieves this objective by adaptively adjusting the knowledge model in the student model and employing appropriate pedagogical strategies to meet individual needs. The details of WAE, the domain model of GWATS and the authoring process to construct GWATS with WAE are covered in Chapter 4. The effectiveness and efficiency of the generic tutoring model and adaptive tutoring strategies are evaluated in Chapter 5.

CHAPTER 4

WEB-BASED AUTHORING ENVIRONMENT (WAE)

WAE is a key component in GWATS and is one of the main objectives of this research. This chapter presents WAE and its implementation details. WAE is implemented in Java Server Page.

According to Murray [102], one of the main goals of the authoring tools is to make it easy and possible for subject domain experts without programming skills to build an educational tutoring system. To achieve this objective, the underlying techniques are employed and hidden as a back-end engine dedicated to dealing with all of the technical problems. Hence, authors without proficient technological knowledge can easily create educational systems for different application domains with the assistance of authoring tools. All of the authoring interfaces and the back-end engine compose the authoring environment. There are two distinct types of users in the GWATS: the ATS authors and the students. ATSs authors customize ATS by using the authoring environment. The generated learning environment is, in turn, used by students to interact with the ATS through the student interface.

One main concern about the ATS authoring tool is what instructional decisions are going to be dealt with by the authoring tool and what will be done by the authors [119]. To relieve the authors of technical tasks and to cope with the individual needs of different ATS authors, the authoring tools have some general functionality that needs special techniques to ease the construction task and limit the authoring scope as domain-dependent matters.

The domain-independent general mechanisms in WAE include representing knowledge in the general structure, using the same topologic structure to model student information and providing the same user interface and the generic back-end tutoring engine. The separation of the knowledge structure from the learning material, student model structure from individual characteristics, the general user interface from the specific learning material presented and the tutoring model from teaching strategies remarkably reduce construction time and energy and make WAE applicable to many subjects. Since the domain structure is separated from the learning material, the authoring interfaces enable authors to create specific learning materials for specific application domains. The authoring tasks left for the author are creation of domain-dependent learning materials and loading the individual characteristics into the student model (Figure 4-1). Then the back-end engine fills the general domain structure with the generated learning contents.

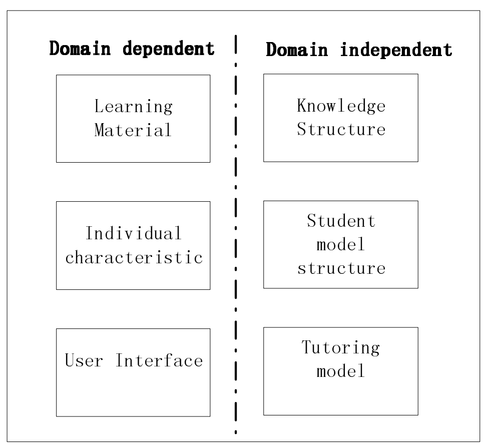


Figure 4-1: Dependent and independent domain mechanisms

Current implementation of WAE is just a prototype system. Its major components are ready to use, but several other components require further development. Authoring ATS using a WAE is a semi-automated process. Based on the domain independent mechanism described above, with the defined knowledge representation and student model representation, the process of authoring ATS can be split into the following steps:

1) Hierarchical domain structure creation, which is currently workable
2) Learning material creation/uploading and associated to related concepts
3) Student static model initialization, which currently is workable but needs further development

The details of the authoring environment and the authoring processes are described in the following subsections.

4.1 Domain Model Authoring

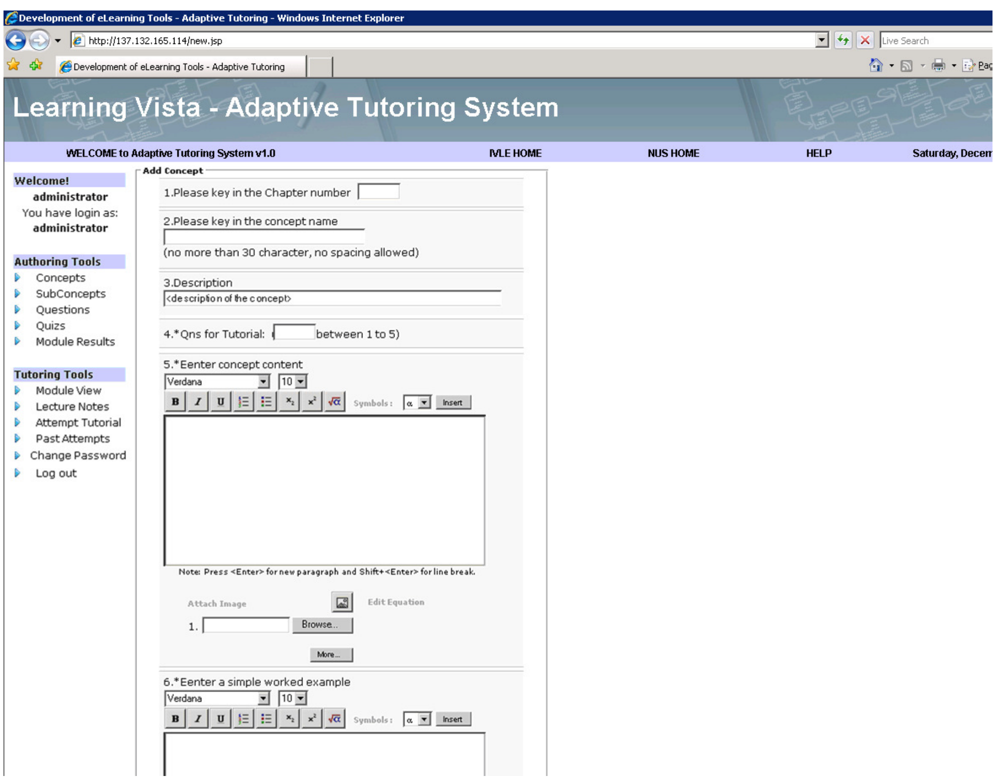


Figure 4-2: Interface of creating a concept

1) **Structuring the domain knowledge**: Concept structure is automatically generated by creating concepts and setting the prerequisite relationships among them. The domain knowledge structuring process starts with concept creation. To create a concept, the lectures should define a set of parameters, such as name, topic, number of questions for assessment and the starting and ending pages of the lecture notes, etc. (Figure 4-2). A concept is an elementary piece of knowledge and is considered the basic unit of knowledge that can be selected or set as the learning objective. Each concept may have multiple prerequisite parents. Concepts are linked by the prerequisite relationships among them, so that a change in one concept may have a chain reaction of impact on all the other linked concepts in the direction of the links. The design of the concept network, in other words deciding which concept is linked to the others, is based on a profound understanding of the applied module. To entertain students with different learning preferences, each concept should be associated with a plurality of learning materials, such as theories or definitions, application examples of the concept, exercises or simulations, questions to assess the concept, etc. The concept addition and edition interface is shown in Figure 4-3. By defining concepts and the dependency between them (Figure 4-4), a concept map will automatically be generated for each subject. Figure 4-4 illustrates an example of a generated concept network.

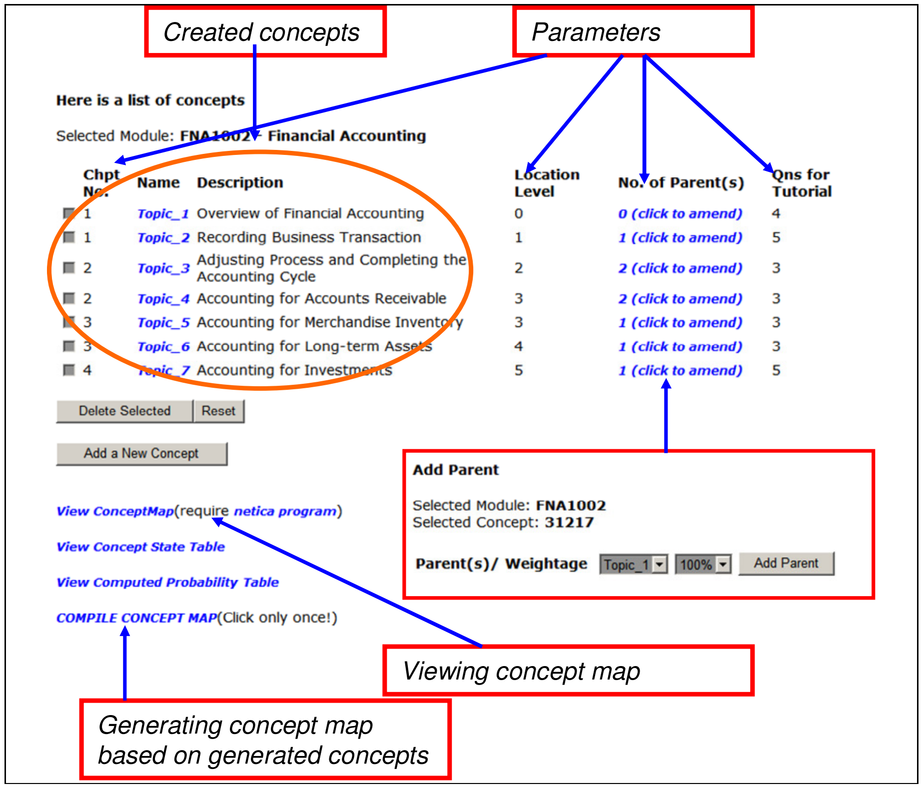


Figure 4-3: **Interface of concept edition**

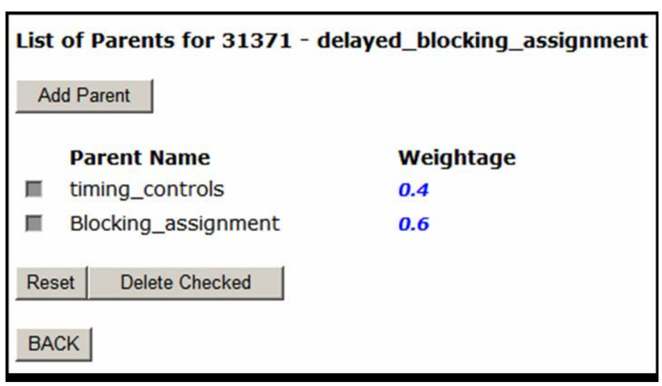


Figure 4-4: **Interface of assigning prerequisite parents and weights**

The concept and concept map creation process requires the lecturer to re-contemplate the cognitive structure of the domain knowledge and reflect the relationships among concepts. Therefore, lecturers can effectively organize the curriculum and can bring forward an efficient method to assess the student's mastery states of the subject.

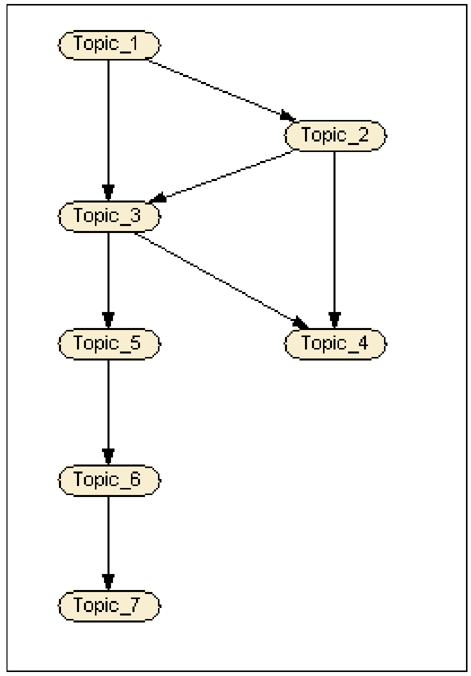


Figure 4-5: **The generated concept network**

2) **Questions creation:** Questions are required to assess the mastery state of the related concepts. The relationships among questions and related concepts are a prerequisite, which means that to correctly answer a specific question, students need to understand all related concepts. Each question could have one or more prerequisite concepts. In WAE, questions can either be created from scratch or uploaded from a database.

To create a question, the lecturer needs to define the question type, the question content, the answer, the prerequisite concept(s), the corresponding weight denoting the importance of each related concept to the question and the hints on different specific levels provided to the student when he or she requests help. To copy existing questions from a database, the lecturer needs to find the appropriate ones by defining its parameters, such as related topics, related concepts, etc., and then copying the selected questions from the source to the destination. Figure 4-6 shows the interfaces of creating new questions and copying existing questions.

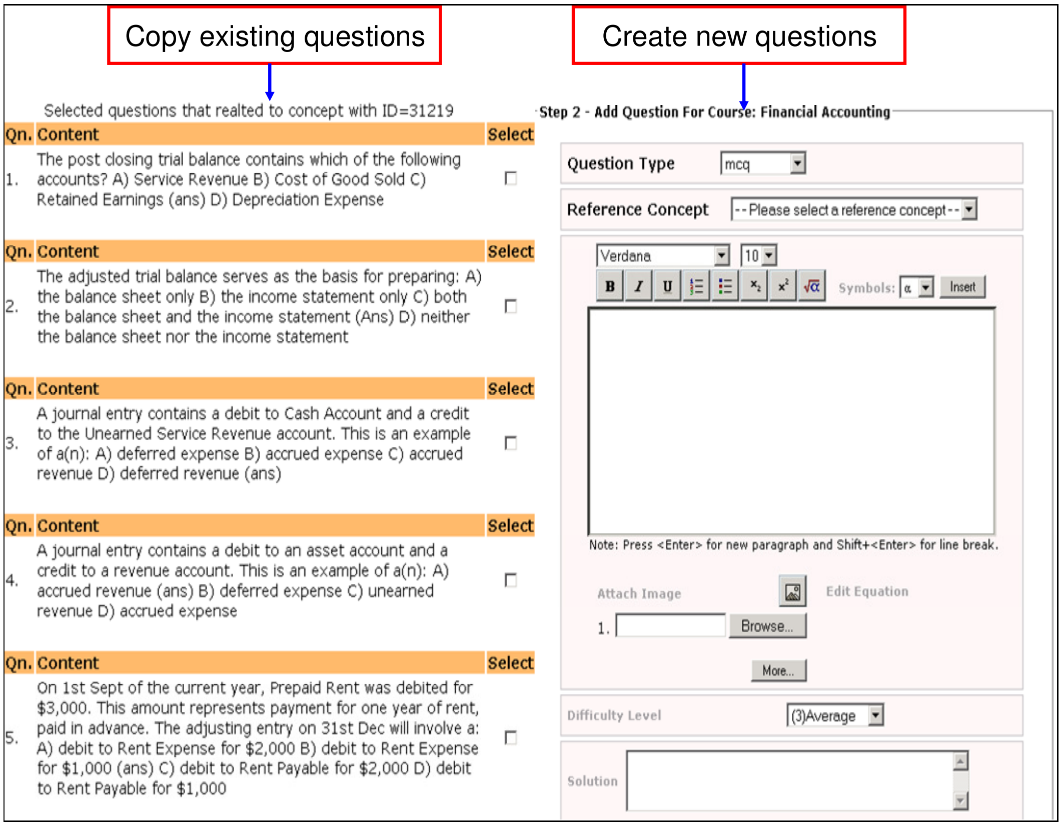


Figure 4-6: **Interface of question creation and edition**

3) **Attaching learning materials to the course structure:** In the current version of WAE, learning materials are actually attached to the course structure during its creation process by defining its prerequisite parents (Figure 4-7), and the corresponding weight, i.e., the importance of the concept to mastering the question. Questions are usually used to assess a student's understanding and mastery states of related concepts.

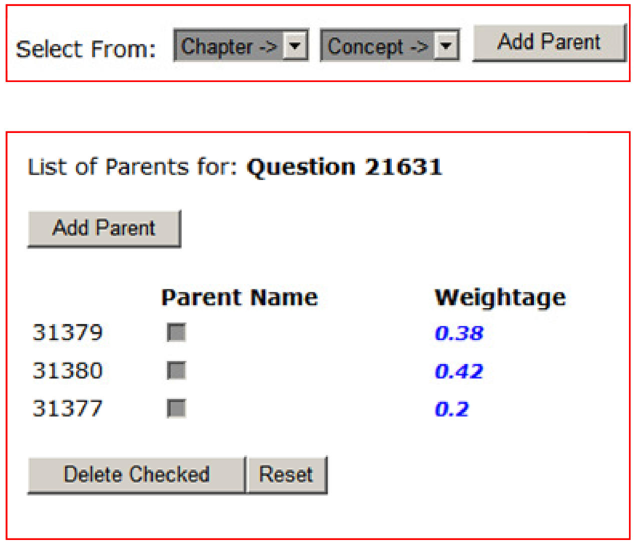


Figure 4-7: **Interface for assigning questions to concepts**

4) **Converting the concept network to a Bayesian network:** Once all of the concepts within the domain model have been created and the prerequisite relationships have been defined, the whole hierarchical domain structure is generated. The concept network constructed during the domain creation can be used as the basis for a Bayesian student model by adding some required parameters (Figure 4-8), because the network structure and representation of the student model at the concept level have exactly the same topology as the concept network. Required parameters are prior probabilities attached to the root concepts nodes representing a statistical measure of students' mastery states of simple concepts, conditional probabilities among simple concepts and their parents' nodes and prerequisite relationships for these variables.

To convert the concept network into a generic Bayesian student concept map, the essential data is as follows:

- Name of concepts (nodes)
- Prerequisite relationships among concepts (links among nodes)

- Prior probability table and conditional probability table

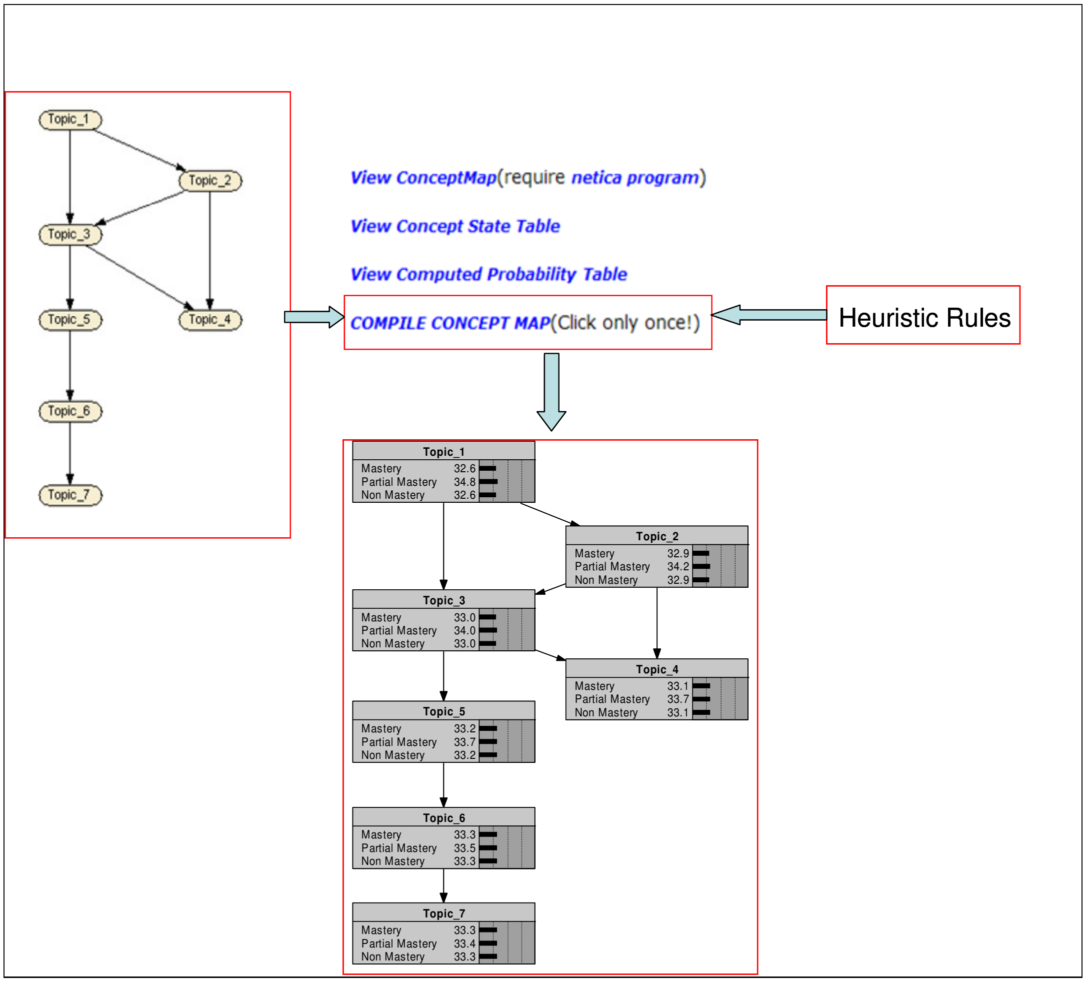


Figure 4-8: **Concept of compiling a concept map into a Bayesian student model**

Names of concepts are entered by the author and prerequisite relationships are defined during the concept creation process. The underlying back-end engine determines the prior probabilities for each concept and the conditional probabilities among them based on the heuristic rules. It affiliates the probabilities to the concepts and the relationships, and then compiles the concept structure into a concept Bayesian network (Figure 4-9). What needs to be mentioned is that the effectiveness of the Bayesian student model is closely related to the Bayesian network design and the accuracy of values entered in the probability tables, including prior and the conditional probabilities.

Getting Bayesian parameters, both prior and conditional probabilities for a certain concept, according to its prerequisite concepts, is a very tough issue. Basically, there are two ways to get these probabilities: set by experienced domain experts in the absence of data or learned from historical data. Before WAE is put into application, it is impossible to get historical data. Although these parameters can be determined by the domain experts based on their experience, if too many parameters are involved, obtaining them by means of expert elicitation is not feasible. Also, expert determination could be too expensive or not accurate enough. GWATS employs an alternative to define these Bayesian parameters.

Prior probability values, representing a randomly drawn student from the population who has already mastered that concept before using the system, might also be derived from empirical data or from the results of a prior test. However, in many cases, it is impossible to get test results or empirical data before the system is authored. Therefore, GWATS assumes each concept has a uniform prior probability distribution—the probabilities of mastery, partial mastery and non-mastery are initially 33%, 34% and 33%. With the ATS generated and put into use in the application domain, the prior probabilities will be updated from the collected data.

Conditional probability tables (CPTs) quantify the probability of a variable in any particular state, given the states of the variables linked to it, i.e., its parental variables. With the Bayesian structure defined, we developed a heuristic algorithm to set conditional probabilities based on the prerequisite relationships and the weights defined by the expert teachers. The heuristic rules and algorithm are a starting point to initialize the parameters and is the measure taken to collect analytical data before accurate and useful results can be obtained. The collected data should be used for the adaptation of the model to the particularities of each student or group of students on a certain domain. This section presents the heuristic rules. The algorithm of learning conditional probability tables from historical cases is discussed in the following subsection.

The heuristic method takes the relative weights of the parent concept to the child concept into consideration when computing the CPTs. This is very natural because the weight symbolizing the importance of the parent-to-child concept. It uses the following rules:

$$probability(Child \mid Parent) = \begin{cases} \sum_{n} w_{p,q} - i * k, \; if \;\; child\_state == parent\_state \\ \sum_{n} k, \; if \;\; child\_state != parent\_state \end{cases}, \qquad (4\text{-}1)$$

where i is the number of states of the concepts, $w_{p,q}$ is the weight of parent node p to the child node q, and $0 <= w_{p,q} <= 1$. n is the number of parents the child has; k is an empirical constant that measures of related uncertainties such as careless errors, lucky guess and change in the student knowledge due to learning and forgetting. In GWATS, i is set to 3 and k is defined by the expert authors depending on the objective student.

5) **Question node CPTs with heuristics:** Questions nodes have two states, true or false, for questions answered correctly or incorrectly. Parents of question nodes are concepts to be understood before correctly answering the questions, which has three mastery states: mastery, partial mastery and non-mastery.

Most Bayesian adaptive testing systems use the IRT logistic function to parameterize the CPTs among test questions and concepts [120, 121]. GWATS employs a generic heuristic algorithm to set the probabilities of each individual student correctly or incorrectly answering the question. The heuristic algorithm takes the importance of the prerequisite concepts to the assessed question and the difficulty level of the question into consideration. The rules were tested with few different sets of parameters to choose the set that gave the best performance. From the experiment, the following rules led to the best results:

$$P(Q_i = True) = \left(\sum_{k=1}^{N} w_{k,i} * p_k\right) - 0.012 * DL$$
$$where,\ p_k = \begin{cases} 0.98, C_k = M \\ 0.76, C_k = PM \\ 0.42, C_k = NM \end{cases}. \tag{4-2}$$

$$P(Q_i = False) = 1 - P(Q_i = True). \tag{4-3}$$

DL is the difficulty level of the question. C_k represents the mastery states of the prerequisite concept C and its values $M / PM / NM$ correspond to mastery/partial mastery/non-mastery, respectively. For example, in the question *Q21436* there are two prerequisite concepts DC_setup with weight 0.6 and Constraint_design with weight 0.4. The difficulty level of the question is 4. The probability conditioned on the mastery states of its two parents are listed in Table 4-1: Initial question-concept CPT set based on heuristic rules.

Table 4-1: Initial question-concept CPT set based on heuristic rules

DC_setup	Constraint_design	P(Q=true)	P(Q=False)
M	M	0.932	0.068
M	PM	0.844	0.156
M	NM	0.708	0.292
PM	M	0.8	0.2
PM	PM	0.712	0.288
PM	NM	0.576	0.424
NM	M	0.596	0.404
NM	PM	0.508	0.492

WAE initially used the generic heuristic rules to initialize the conditional probabilities of the question according to the mastery states of the prerequisite concepts. Once the system put into application, with learning cases and empirical data gradually collected from students, GWATS enables authors to amend the CPT for a particular student group. Table 4-2: Revised question-concept CPT based on the collected learning cases lists the CPT revised by collected historical data. As more data or knowledge becomes available, it may be necessary to revise these parameters to reflect the improved data set.

Table 4-2: Revised question-concept CPT based on the collected learning cases

DC_setup	Constraint_design	$P(Q$=true)	$P(Q$=False)
M	M	0.875	0.125
M	PM	0.7	0.3
M	NM	0.425	0.575
PM	M	0.667	0.333
PM	PM	0.572	0.428
PM	NM	0.402	0.598
NM	M	0.395	0.605
NM	PM	0.375	0.625
NM	NM	0.364	0.636

4.2 Student Model Authoring

In Chapter 3, the student knowledge model in GWATS consisted of static and dynamic parts. Accordingly, the student knowledge model authoring includes two steps.

1) **Static student modeling authoring:** A student static model is generated from the concept network of the module. Each module has one static Bayesian concept network. With the prerequisite relationships and the weights among concepts, it is very convenient to convert the concept network into a Bayesian network. The key to the conversion is to derive the Bayesian parameters within the defined structure.

When the student registers with the module for the first time, the system copies the module Bayesian concept network (Figure 4-9) as the initial states of the student. This research assumes that every student has the same initial static student model. As the student progresses through tutorials, the system collects personalized evidence and injects individual learning states, characteristics and performance into the individualized student model.

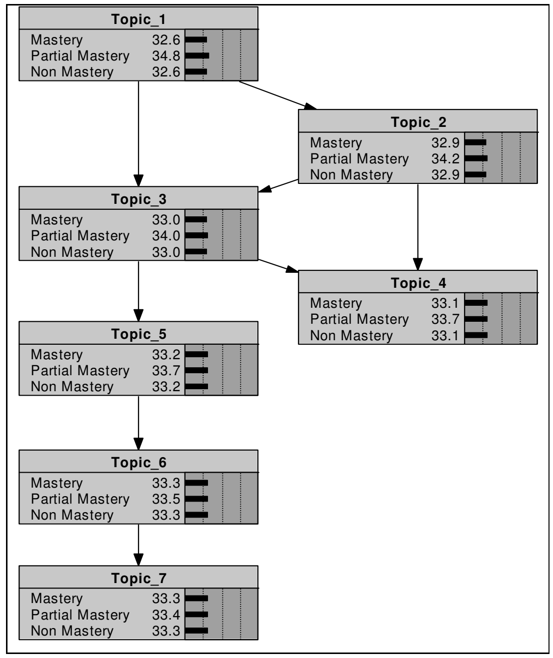


Figure 4-9: **An example of a static student model**

2) **Algorithm to learn CPTs from case files:** As mentioned, the effectiveness of the student model is closely related to the Bayesian network design and the accuracy of the values entered in the probability tables. The amount of data available for collection after the system is put into use makes parameter learning a valid option for this project. We used the EM (expectation-maximization) algorithm, one of the most common Bayesian parameter learning methods, to find the conditional probabilities from the specific historical data based on the defined structure. With the learning cases accumulated, the conditional probabilities became more accurate and the system indicated better decisions.

Many real problems are faced in a situation in which only a subset of a given Bayesian network's variables are observable from the data. This seems inevitable for learning CPT from accumulated learning cases. This is because learning cases from the student are only partially observed. Only the related concepts' mastery states are observed. The mastery states of non-related concepts are unobservable. The EM algorithm can be used to learn a Bayesian network's parameters in the presence of hidden variables. In this case, although their values are not observed, the EM algorithm determines that they exist and must find a place for them in the network.

The basic idea of the EM involves two steps [122, 123]: the E-step (expectation step) and the M-step (maximization step). Assume that we have some incomplete data X that is described by a set of parameters θ. The EM algorithm is an iterative procedure that improves the data log-likelihood $\log(P(X \mid \theta))$ at each step. If Z is the set of missing data, the parameters θ_{i+1} (at iteration i+1) are estimated from parameters θ_i (at iteration i) using the following formula:

$$\theta_{t+1} = \arg\max_{\theta} E_{P(Z \mid X, \theta_t)}[\log P(X, Z \mid \theta)] \tag{4-4}$$

The E-step computes $P(X, Z \mid \theta)$, the posterior probability of missing data given observed data and current parameter estimates. The M-step re-estimates the parameters by maximizing $\log P(X, Z \mid \theta)$, the expected log-likelihood of complete data, assuming the missing data comes from the distribution computed in the E-step.

The E-step and M-step constitute an iterated loop. Each E-step finds expected values for the hidden variables that better fit the data than the previous iteration. Each M-step involves finding improved parameters that better fit the data than the previous iteration

[124]. Under certain circumstances, the algorithm has been shown to converge to a local maximum likelihood hypothesis [125], and then we can get the conditional probability parameters of the static student model.

The CPT learning algorithm is offline learning and runs in the back-end. When the lecturer presses “Learn question CPT from cases,” the system automatically reads unused cases from the database. The used cases will be marked or deleted from the database. After the system gets the new CPT from learning cases, the student model should be updated with the new CPT. For our project, Netica API was used for network manipulation and inference and for calculating CPT parameters from case files. Netica API assumes the conditional probabilities being learned are independent and the prior distribution is Dirichlet, with which our system is totally satisfied. Netica API implements the Expectation Maximization (EM) algorithm for parameter estimation [124]. This algorithm allows the presence of missing values, generates presentation quality graphics and has functions for easily performing statistical tests and sensitivity measurements on belief networks using test data.

Table 4-3: Initial concept-concept CPT set based on heuristic rules

Data Types	**Parameters**		
	Mastery	Partial Mastery	Non-Mastery
Mastery	0.99	0.005	0.005
Partial Mastery	0.005	0.99	0.005
Non-Mastery	0.005	0.005	0.99

The initial CPT of mastering the destination concept “Parameters” given its parent “Data Type” is set based on heuristic rules (Table 4-3). The CPT learned from the historical data is shown in Table 4-4:

Table 4-4: Revised concept-concept CPT learned from the historical data

Data Types	**Parameters**		
	Mastery	Partial Mastery	Non-Mastery
Mastery	0. 85714	0.07413	0.07413
Partial Mastery	0.12	0.78	0.1
Non-Mastery	0.08	0.25	0.67

As described above, the student have the same initial CPT. With learning cases accumulated and used as historical data to revise the CPT, the collected learning cases represent the learning characteristics of the person who uses the system. So if the GWATS is devoted to an individual student, then the system is gradually injected into the individual knowledge information and personality. For the system used by a group of students, a CPT improved by the historical data can more accurately represent the overall performance of the group along with the wider use. The CPT revising process cannot be accomplished in one action. It is a long-term, step-step process of the GWATS evolving from being generic to being individualized. The longer time used, the more learning cases collected and the more individual information gathered, the better the performance.

3) **Dynamic student model generation:** This model is automatically generated every time the student selects to attempt a tutorial. After the student chooses a target concept to attempt in a tutorial, the tutoring model selects appropriate questions using the

questions selection method described in Chapter 3. Then it creates a Bayesian network with the selected questions and the related concepts (Figure 4-10). All the parameters of the concept nodes are directly copied from the static student model that represents a student's mastery states of the related concepts over time. The conditional probabilities of the question, given its parents concepts, are calculated based on the weights from each of its parents.

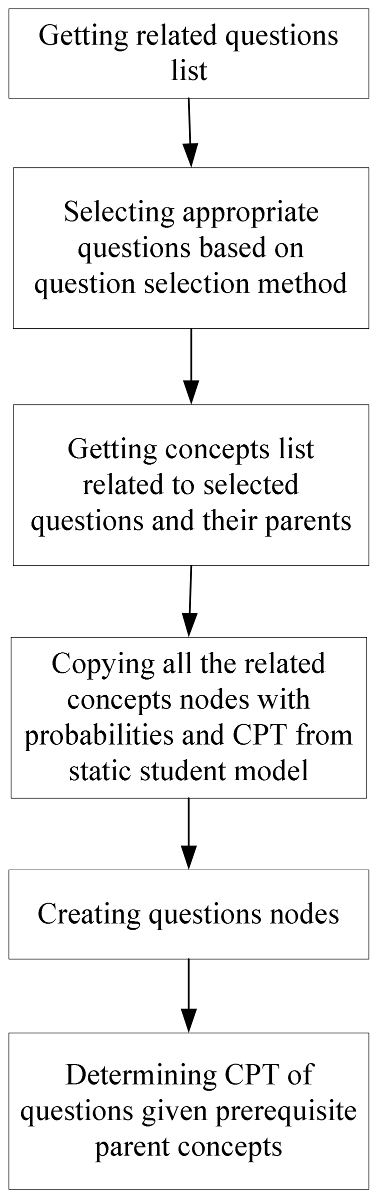


Figure 4-10: **Procedure for dynamic student authoring**

The dynamic student model is mainly used to infer the student's mastery states of the related concepts. It observes the student's responses to questions and updates the probabilities. Because the probability of the student solving the question correctly depends on the knowledge of the associated concepts, the correctness/incorrectness of the questions allows us to make inferences about his or her knowledge level of the related concepts. Similarly, the probability of a student mastering one concept tells us some information about all its prerequisite concepts. Therefore, the updated probabilities and mastery states of the concepts in the dynamic student model are considered as collected evidences to update the static student model. Figure 4-11 shows one example of generated dynamic student model.

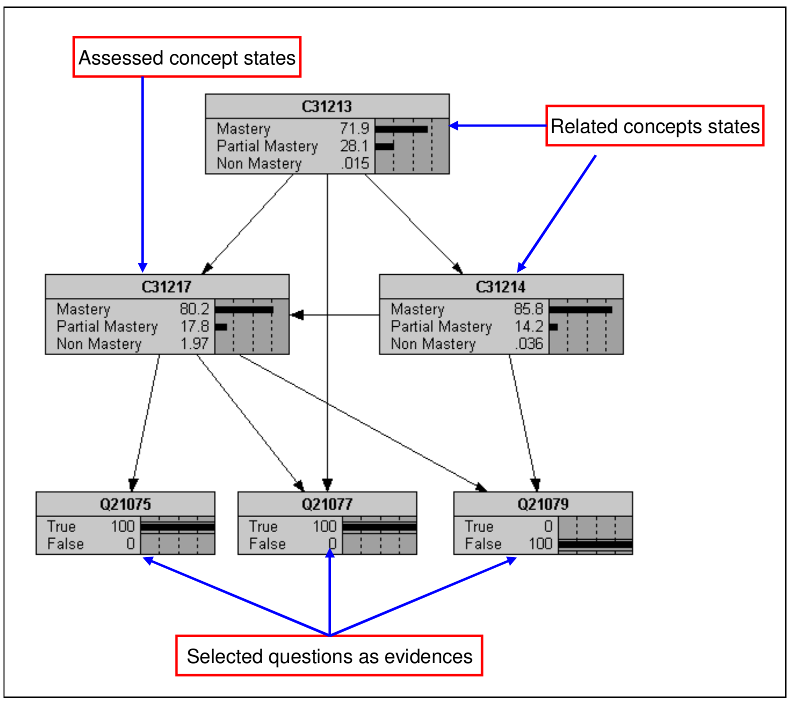


Figure 4-11: **Example of generated dynamic student model**

4.3 Student Interface

Authoring tools are the interface for the authors and are described in the domain authoring section. This section will show the adaptive learning environment that GWATS provides for students. Learning starts with choosing a learning goal. Students can set whatever he or she wants as a learning objective.

GWATS accommodates students with different learning styles and stimulates them to learn through Kolb's experiential learning cycle through multiple types of educational materials (Figure 4-12). After a learning goal is chosen, the adaptive tutoring system presents all types of the available learning materials related to the learning objective to the students, so they get their favorite learning materials, which include:

1) Multimedia demonstrations for students to access from anywhere at anytime.
2) Visualization tools to help students understand the difficult concepts.
3) Online simulations for practicing the theories learned in the class.
4) Virtual laboratory to enhance the understanding of concepts.
5) Adaptive tutorial system to accurately estimate students' mastery states.
6) Adaptive tutorial systems to provide appropriate tutorial questions that are neither too easy nor too difficult.
7) Adaptive tutorial system to offer meaningful feedback to enhance the learning efficiency.

Meanwhile, the student interface tailors an interactive and hands-on learning environment to help students understand and it facilitates active learning. But students are qualified to attempt the target only if he or she performs sufficiently well to show that he or she has gained the knowledge of all of the prerequisite concepts. Tutorial questions are selected by a tutoring model based on each student's learning status and learning history, which exemplifies the idea of individualized tutoring.

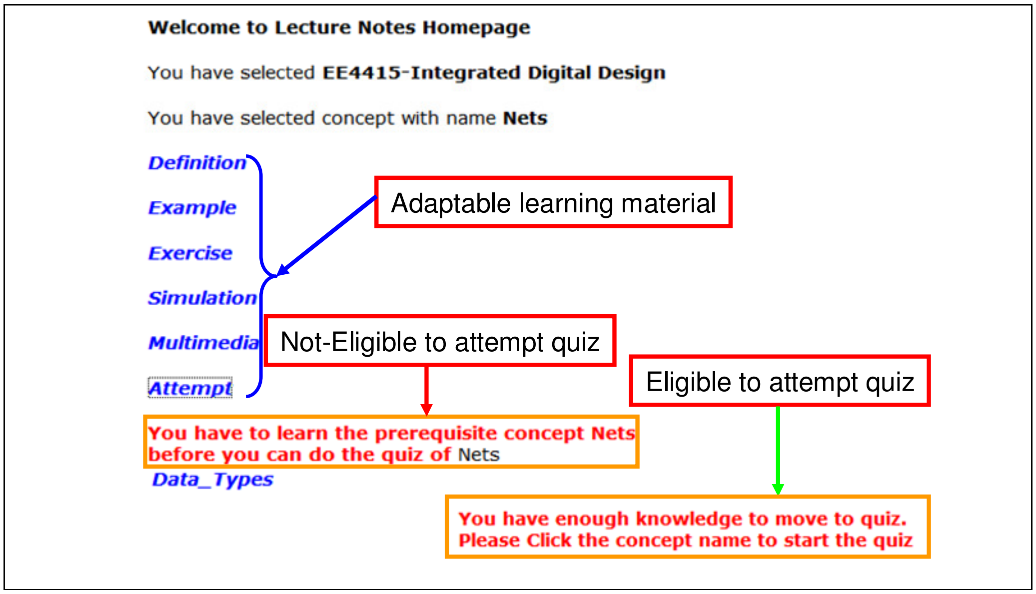


Figure 4-12: **Student learning environment**

4.4 Quantitative Evaluation

We created a web-based adaptive tutoring system using WAE in an engineering subject: Integrated Digital Design with a module code: EE4415. WAE is friendly to users and most of its authoring tools are convenient. It takes less than 90 minutes to train authors. The most basic, but the most important, part in creating an ATS using WAE is to create the domain structure (the concept map). The effectiveness of the ATS mostly depends on the accuracy of the concept map. There are three steps to create a domain concept map: creating all the key concepts or topics in the domain; defining the relationships among them by setting their prerequisite concepts and the importance of each prerequisite concept to the target concept; and generating the domain concept map with the back-end engine, based on all the node concepts and parameters entered for the domain. For an expert in the domain who knows the key concepts and their relationships very well, it took about 100 minutes to create and generate the whole concept map with 36 concepts.

The majority of time was spent on loading the concept-related information—concept definition, examples, exercises, simulations, visualizations, virtual labs and tutorial questions—to the database. WAE provides uploading tools for instructors to integrate existing learning materials into the ATS. It saves time and energy and spares authors from creating every question from scratch. For 36 concepts, creating 72 exercises, 72 examples and 144 tutorial questions from scratch and uploading existing simulation and visualization materials took us 96 hours to complete. As described, the initialization of the student model was very simple, taking less than two minutes. Because the current WAE uses a generic tutoring strategy to provide adaptive tutoring, once the student model is initialized and learning material is generated, all the authoring work is done.

Authoring tools reduce the time and cost required to build the instructional systems or, with the same amount of time and effort, to improve or increase the effectiveness of instructional systems [102]. The effectiveness of the ATS created by WAE is not very easy to evaluate and needs a complicated experiment to compare and analyze the learning outcomes (see Chapter 5). The effectiveness of the WAE related to the effort needed to create instructions can be calculated by examining the effort required to create the ATS system, which was 5,860 minutes. This total included 100 minutes to create the concept network, 96 hours for related learning materials and the time taken by the system to generate all online instructions. The ATS for EE4415 includes 36 concepts, made up of 16 easy, 12 intermediate and 8 difficult. The required time for each concept is about 15 minutes, easy; 20 minutes, intermediate; and 25 minutes, difficult. The overall online instruction time was 680 minutes. Therefore, the approximate ratio of 5,860 minutes of authoring for 680 minutes of instruction is 8.62 to 1, which is quite favorable compared with 300:1 for the traditional CAI [102]. However, this ratio is for the case that we upload all the simulations, visualizations and virtual labs, without counting the effort for creating these multimedia learning materials. Creating all learning materials from scratch would take more time.

There are relatively few evaluations of ITS authoring tools, partly because there are too many features to evaluate and it is difficult to measure the effect of each feature one by one. In this chapter, we used WAE to create a web-based adaptive tutoring system, put it into application and demonstrated its efficiency by evaluations. But constructing an ATS using authoring tool only demonstrates the usefulness of WAE and is far from sufficient to prove the effectiveness of GWATS. Chapter 5 evaluates and analyzes the effectiveness of GWATS related to the students' performance or learning outcomes.

CHAPTER 5

THE EVALUATION OF GWATS

GWATS is a solution to promote individual learning and is an educational system widely employed for different subjects. It promotes individual learning by providing an ATS that adapts to each student's learning preferences and learning status. It promotes an educational system widely employed by providing authoring tools to ease the construction of the ATS. The design considerations, architecture and the key components and characteristics of GWATS were described in Chapter 3. We presented the WAE and described its architecture, key components and authoring process in the Chapter 4. WAE has been used in creating ATSs or several subjects, like financial accounting, and Digital Signal Processing. However, the "existence proof" of the authoring tools that have been successfully used to produce a variety of ATSs is not enough [102]. We need a number of qualitative and formative evaluation methods to ensure that WAE is usable, friendly and effective. The efficiency and effectiveness of the adaptive technologies that GWATS employs are also presented in this chapter.

5.1 Introduction

Evaluation is the obligatory part of the research. The usefulness and effectiveness of any research can only be determined with evaluations. For our research, the usefulness and effectiveness of GWATS is evaluated in a series of studies that focus on different aspects of the system.

The tutoring model plays a central role in GWATS. We conducted an experiment to validate the usefulness and effectiveness of the adaptive features, like adaptive concept selection method, the prerequisite filtered method and adaptive question selection method. Both simulated students and the real students were used to evaluate the effectiveness of the tutoring model and the GWATS. The validity of the tutoring model has been tested by using simulated students. The results obtained show that the Bayesian tutoring strategies have an excellent performance in terms of accuracy and that the introduction of adaptive concept selection and question selection methods improve its behavior in terms of accuracy and efficiency. Real students were used to show that the ATS created by WAE is educationally effective in comparison with traditional instructional methods. It revealed that the tutoring model does facilitate the personalized learning.

In addition, a comprehensive survey evaluation was carried out with three groups of students tutored with different sets of teaching strategies to evaluate the adaptive features and the effectiveness of the system. The evaluation shows that GWATS was friendly and favorable to use. The remainder of this chapter contains details of the three evaluation studies.

5.2 Evaluation with Simulated Students

The objective of this evaluation was to find out whether the use of adaptive concept selection and adaptive question selection methods improved the diagnosis process and promoted students' learning outcomes. We employed a Bayesian network-based simulation environment in evaluating the effectiveness and accuracy of different approaches for concept selection and question selection. Using simulated students in intelligent tutoring systems is not new to the research community [149-153]. The main applications of simulated students have been identified in [132]. There are tutor-training systems, in which a trainee teacher is evaluated while coaching a simulated student [154]. There are collaborative learning systems, in which a simulated student can act as a learning companion for a human student [155, 156]. Finally, there are applications concerned with evaluating tutoring systems [139, 157]. Although the simulated students may not closely mimic human behavior, it will be clear shortly that the Bayesian network-based models offer a convenient infrastructure for capturing the fuzziness in students' responses to test items and the dependent relationships among the test items.

The main reasons for using simulated students are as follows. It does not seem appropriate for testing an evaluation method with real students without proving its validity beforehand. Simulated students are an effective way to test the effectiveness and validity of the algorithm before incurring the cost of real students. Simulated students eliminate other variables, such as the accuracy of other component of tutoring system, providing results that focus exclusively on the performance of the tutoring model. By using simulated students, the cognitive state obtained as the result of the tutoring algorithm can be compared to the student's true cognitive state. The final reason we used simulated students is that these can cover a wide spectrum of abilities and responses effectively and efficiently [132,133].

The effectiveness of the Bayesian student model, the adaptive concept selection and question selection methods are evaluated by calculating the number of concepts that have been correctly diagnosed, incorrectly diagnosed and undiagnosed. To evaluate simulated students, the simulated ATS should be created beforehand. Simulated students in GWATS allow us to do an evaluation based on a Bayesian concept network and the question-concept relationships without defining real course content and questions for a particular subject.

5.2.1 Introduction about the Experiment

In the following section, we describe how to create the simulated ATS and how a simulated student is generated.

a. *The design of simulated domain model.* The key step to construct an ATS based on WAE is to build the domain model and create the Bayesian concept network. Before creating simulated students, the domain model needs to be designed. We created a concept network with 12 concepts and 180 questions, as shown in Figure 5-1. Each question is related to one to three concepts, and each concept is related to several questions. In this concept network, concepts C1, C2 and C3 are considered to be "easy" since they have no prerequisite concept, whereas C10, C11 and C12 are considered as "very hard."

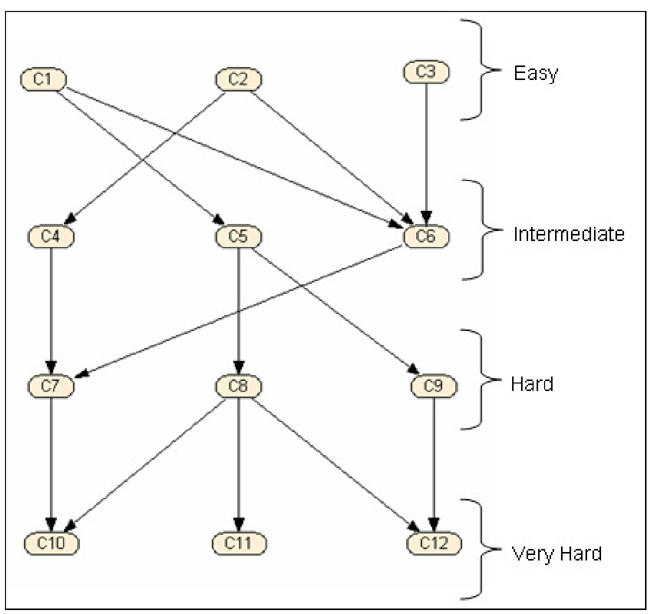


Figure 5-1: **Concept network for simulation module**

b. *Simulated students groups.* The students were put into four categories, according to their knowledge states on all concepts: novice, intermediate, good and expert (Table 5-1). Simulated students belonging to the same group have the same knowledge level, but might know different sets of concepts.

Table 5-1: Known and unknown concepts for each category of students

Student Categories	Known concepts (Fixed)	Known concepts (Variable)
Novice	–	C1-C3
Intermediate	C1-C3	C4-C6
Good	C1-C6	C7-C9
Expert	C1-C9	C10-C12

c. *Generating simulated students.* We compiled the concept network of the simulated domain in the first step with the prerequisite relationships and the weights among concepts to create a generic Bayesian student model. Each simulated student is assumed to be "known" or "unknown" regarding a concept. Different students have different sets of "known" concepts. Simulated student models are generated by initializing the generic student model with the prior mastery states of each simulated student over all the concepts. Sixty simulated students were randomly generated for each type by the simulation program, taking into account the difficulty of the concepts and consistency with prerequisite relationships. The set of concepts in the second column of the Table 5-1 are known by all the students in that category. To determine whether the concepts in the third column are known, a random number was generated for each concept. If the generated number was larger than 0.5, the concept was taken as "known." Along with this way, each student had a different set of known concepts, although they belong to the same category. The evaluation with the generated students was carried out using the procedure as shown in Figure 5-2, which is discussed in more detail in following sections.

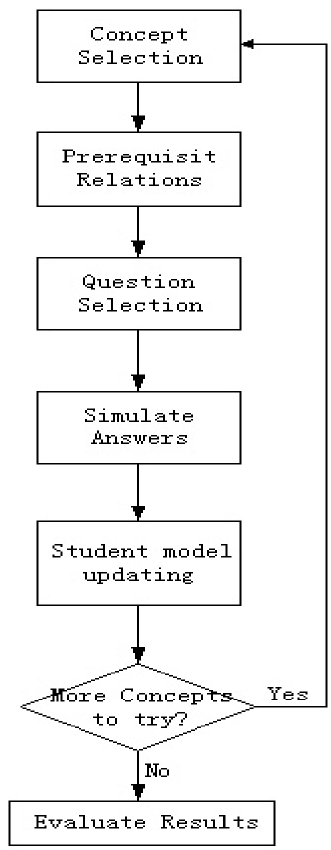


Figure 5-2: **Procedure of the experiment**

d. *Adaptive concept selection evaluation.* The adaptive concept selection method of ATS was evaluated against sequential concept selection. The adaptive method selects the concept for which prerequisite concepts are all mastered as the target concept for the student to learn. However, in sequential concept selection, each concept is targeted exactly once regardless of the student's concept states, depending on the location level of the concept in the concept map. The concept with a lower location level was selected before the ones with higher location levels. In this way, the number of questions that satisfy the prerequisites for higher location level concepts was increased, compared with a random concept selection.

e. *Filtering questions with prerequisite relations.* After the target concept was determined, questions related to the target concept were selected and presented to assess whether the student mastered them. There were two steps: first, questions were filtered with related concepts, and a set of questions was selected based on certain criterion. There were also two ways to filter questions. One was selecting questions related to the target concept regardless of what the knowledge states of other related concepts are. The other method was employed in ATS, which took the prerequisite relationships into account and only questions that satisfy the prerequisites were selected. That is, except for the target concept, all the other parent concepts of the question had to be mastered or partially mastered for that question eligible to be selected. Prerequisite relations can be obtained from the Bayesian map of the individual student. The prerequisite relation filtering method was evaluated through comparisons with the one without prerequisite relations.

f. *Question selection.* The effectiveness of the information gain selection method and conditional probabilities selection method were evaluated by comparing them to the random question selection method, which randomly selected questions related to the target concept regardless of the mastery states of other relevant concepts.

g. *Simulating answers.* Once a set of questions were selected for a tutorial, answers were simulated depending on the knowledge states of each student. For each question, a random number r was generated. The probability of the student getting that question correct p was calculated with his or her Bayesian student model. If

p was larger than or equal to r, the student's answer to that question was considered correct. Otherwise, it was deemed incorrect.

h. *Evaluation.* It is supposed that there is no "learning" taking place during the assessment process. The criteria used to determine whether the concept was correctly or incorrectly diagnosed are as follows: A threshold value is needed to determine the mastery states of the student.

- A concept was **correctly diagnosed** if the simulated student knew the concept but was diagnosed as known or if the simulated student did not know the concept but was diagnosed as unknown, which means that: *(Knowledge == known) AND (Mastery state>= threshold) Or (Knowledge == unknown) AND (Non-Mastery state>= threshold).*
- A concept was **incorrectly diagnosed** if the simulated student knew the concept, but was diagnosed as unknown or if the simulated student did not know the concept but was diagnosed as known, which means that: *(Knowledge == known) AND (Non-Mastery state>= threshold) Or (Knowledge == unknown) AND (Mastery state>= threshold).*
- A concept was **undiagnosed** if its mastery probabilities was lower than the threshold, which means that: (*Knowledge == known) AND (Non-Mastery state< threshold) AND (Mastery state < threshold)) OR ((Knowledge == unknown) AND (Non-Mastery state< threshold) AND (Mastery state < threshold)).*

5.2.2 Experiment and Results Analysis

Once simulated students were generated, we used them to carry out evaluations by using different strategies in concept selection or question selection step, like the effectiveness of prerequisite relationship, the efficiency of the adaptive concept selection strategy and the efficiency of the adaptive concept selection methods.

a. To evaluate the effectiveness of the prerequisite relationships, two sets of simulated students were used. One considers prerequisite relations and selects questions satisfying the prerequisite relations. The other did not consider prerequisite relations. The sequential concept selection method was used in this experiment for both groups of simulated students. Each took 12 tutorials containing 4 questions. Table 5-2 shows the percentage of concepts diagnosed correctly, incorrectly and undiagnosed for both conditions.

Table 5-2: Evaluation results for the filtered method

Categories	Without Prerequisites	With Prerequisites
Correctly diagnosed	56.9%	85.8%
Incorrectly diagnosed	12.6%	8.5%
Undiagnosed	30.5%	5.7%
Expert	$C1 - C9$	$C10 - C12$

Table 5-3: Breakdown of the number of concepts by with and without prerequisite relations

	Without prerequisite relations			With prerequisite relations		
Concept	Correctly Diagnosed	Incorrectly Diagnosed	Un-diagnosed	Correctly Diagnosed	Incorrectly Diagnosed	Un-diagnosed
C1	64	23	33	96	15	9
C2	69	6	45	108	3	9
C3	82	4	34	117	2	1
C4	54	12	54	110	5	5
C5	68	26	26	91	20	9
C6	48	8	64	95	3	22

C7	56	8	56	112	8	0
C8	75	22	23	109	9	2
C9	79	18	23	97	17	6
C10	53	22	45	107	12	1
C11	87	14	19	106	14	0
C12	85	18	17	87	15	18
Total	820	181	439	1235	123	82
%	56.9%	12.6%	30.5%	85.8%	8.5%	5.7%

As can be seen from the results, the percentage of the concepts that were correctly diagnosed increased greatly after the addition of the prerequisite module. This result can be explained as follows. With the prerequisite module, only the questions with all the parent concepts (except the targeted concept) mastered were chosen. The student will have a higher probability of getting these questions mastered than questions with few un-mastered parent concepts. As shown in Figure 5-3, the number of concepts correctly diagnosed by the ATS with the prerequisite relations was always higher than the ones without the prerequisite relations.

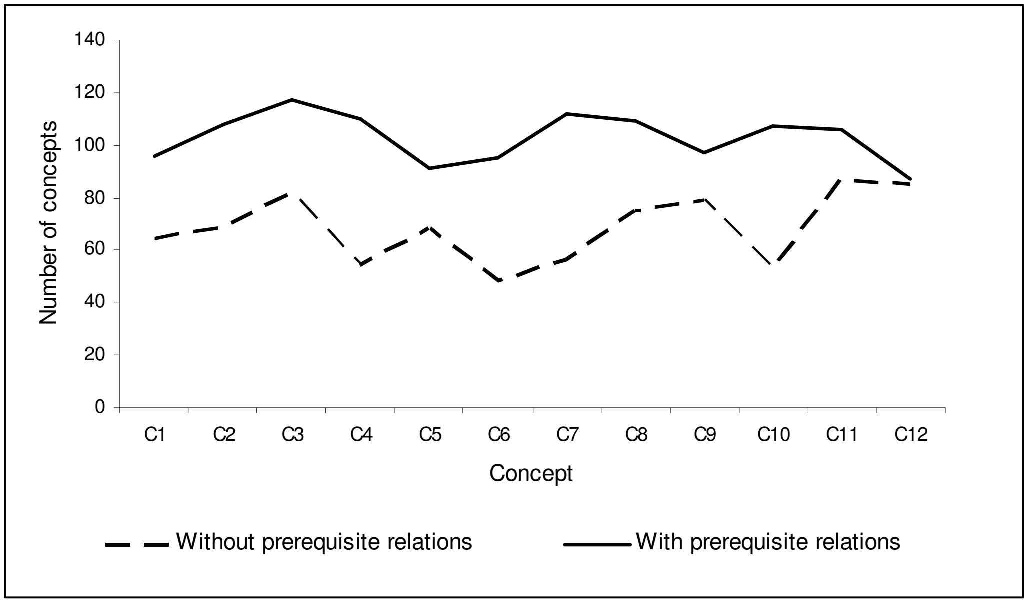


Figure 5-3: Graph of number of concepts correctly diagnosed with and without prerequisites

b. To evaluate the efficiency of the adaptive concept selection strategy, we ran two programs using random concept selection and adaptive concept selection in the concept selection procedure. By random concept selection, we mean that the program randomly selects a concept to assess the student regardless of the mastery states of the concept and its parents in the student model. For the adaptive concept selection, the program takes into account the mastery states of the concepts and selects the most appropriate concept, the parents of which are all mastered or at least partially mastered, to present to the student. The evaluation results are listed in

c.

d. Table **5-4**, which shows that the adaptive concept selection method yields a high percentage of accuracy. Table 5-5 shows the breakdown of the number of concepts for the two concept selection methods.

Table 5-4: Evaluation results for the concept selection method

Categories	Sequential concept Selection	Adaptive concept selection
Correctly diagnosed	85.8%	94.2%
Incorrectly diagnosed	8.5%	4.3%
Undiagnosed	5.7%	1.5%

Table 5-5: Breakdown of the number of concepts by concept for sequential and adaptive concept selection methods

	Sequential Concept Selection			Adaptive Concept Selection			
Concept	Correctly Diagnosed	Incorrectly Diagnosed	Un-diagnosed	Correctly Diagnosed	Incorrectly Diagnosed	Un-diagnosed	Un-attempted
C1	96	15	9	116	4	0	0
C2	108	3	9	119	1	0	0
C3	117	2	1	116	4	0	0
C4	110	5	5	99	7	3	11
C5	91	20	9	105	4	0	11
C6	95	3	22	92	1	2	25
C7	112	8	0	61	3	1	55
C8	109	9	2	67	7	0	46
C9	97	17	6	68	4	2	46
C10	107	12	1	33	2	4	81
C11	106	14	0	49	2	3	66
C12	87	15	18	39	5	0	76
Total	1235	123	82	964	44	15	417
%	85.8%	8.5%	5.7%	94.2%	4.3%	1.5%	

Figure 5-4 shows the percentage of correctly diagnosed concepts for the sequential and adaptive methods; the adaptive method has a higher percentage of correctly diagnosed concepts.

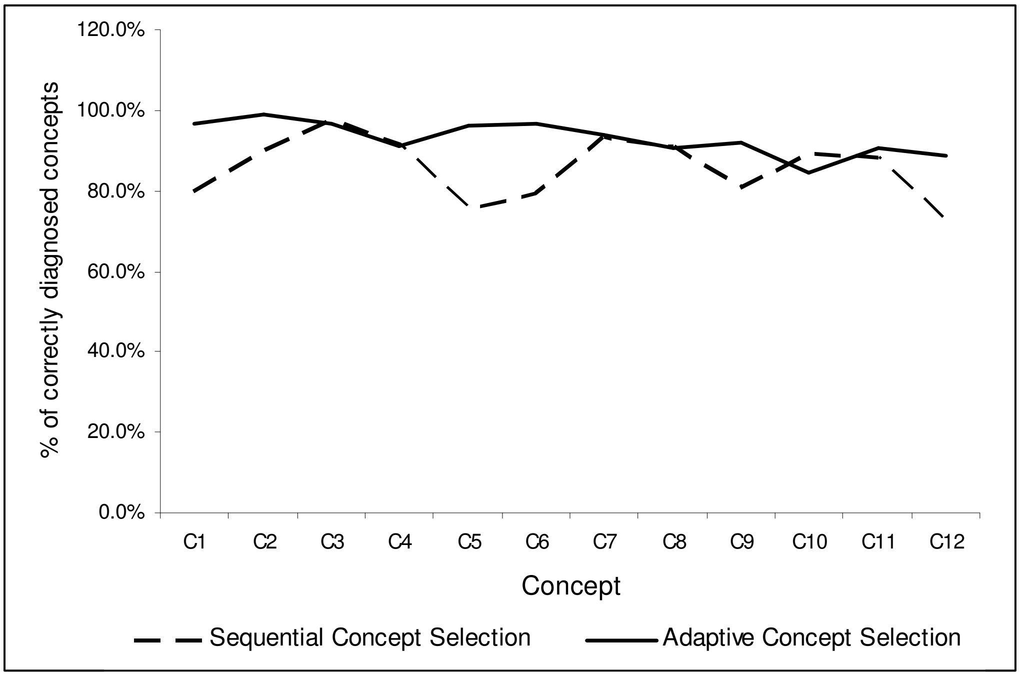


Figure 5-4: The percentage of correctly diagnosed concepts for sequential and adaptive concept selection methods

Figure 5-5 shows the number of undiagnosed concepts for each type of student with the adaptive concept selection method. As shown, most of the difficult concepts are undiagnosed by the novice and intermediate students, whereas most of the concepts are diagnosed by the expert students. From these results, the concept selection algorithm clearly adapts according to the mastery level of each student.

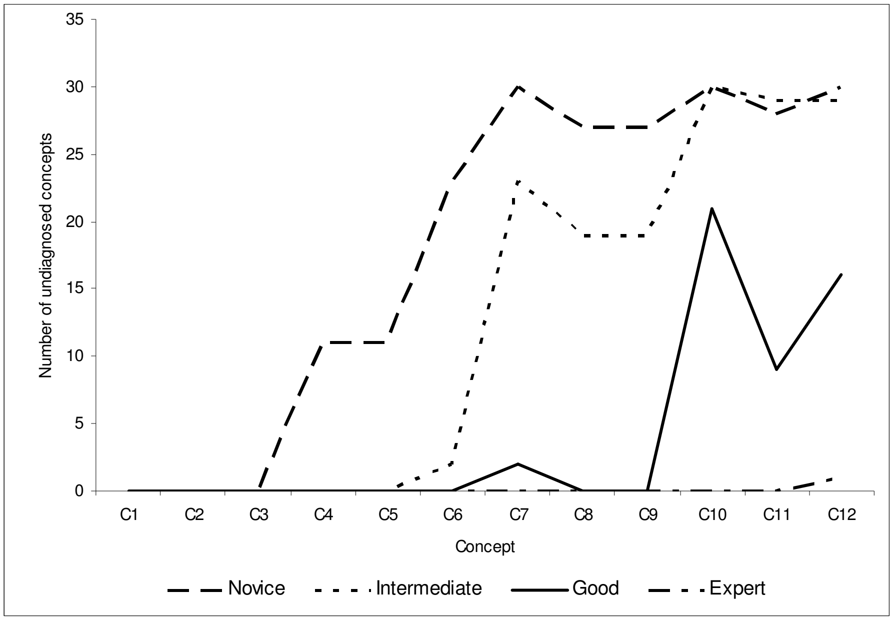


Figure 5-5: Number of undiagnosed concepts of different student types with adaptive concept selection

e. To evaluate the question selection methods, we used three programs having the same adaptive concept selection method followed by a different method for question selection: 1) random question selection: the program randomly selects questions related to the target concept and presents them to the students regardless of the mastery states of all the other related concepts; 2) information gain method: the program chooses questions that maximize the expected reduction of entropy of the test based on the measure of information; and 3) conditional probability method: the program selects questions based on the conditional probability of mastering target concepts if the chosen question is answered correctly. The evaluation results are listed in Table 5-6. The results show that there is no significant difference in the results for the adaptive question selection method compared with the random question selection method. The good behavior of the random method might be attributable to the sound theoretical model of the Bayesian network, which shows an excellent performance in classification and diagnosis problems.

Table 5-6: Evaluation results for the question selection method

Categories	Random question selection	Information Gain Method	Conditional Probability Method
Correctly diagnosed	94.2%	94.8%	94.6%
Incorrectly diagnosed	4.3%	5.2%	3.4%
Undiagnosed	1.5%	0.0%	2.0%

Figure 5-6 shows the number of questions diagnosed correctly for the information gain method and the random question selection method for each type of students. There is no significant difference in these methods for any type of student.

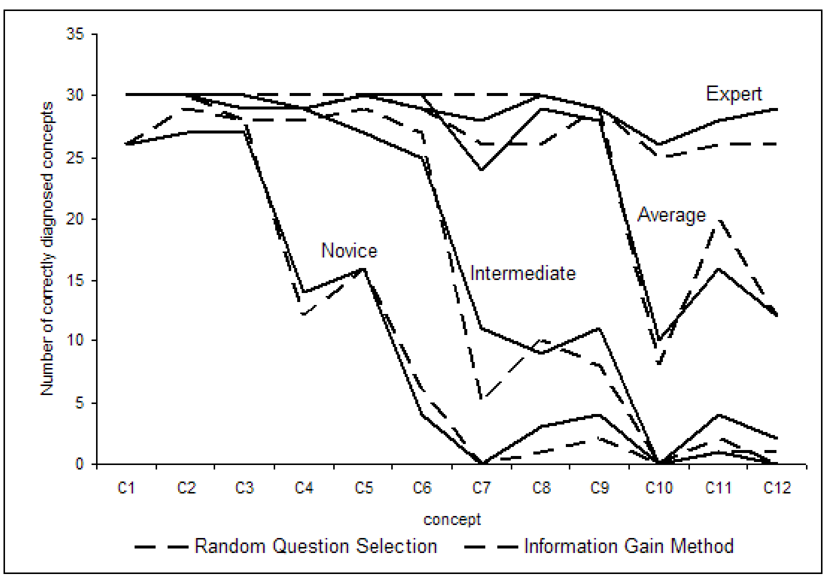


Figure 5-6: Number of correctly diagnosed concepts by type of students using random and information gain question selection methods

The number of correctly diagnosed concepts decreased for some of the concepts, for example, C7 and C10. This can be explained by the number of prerequisite parent concepts. This number can be derived from the concept map of the course shows the number for each concept. From this table, C7 and C10 have the highest number of prerequisite concepts. Therefore, these two concepts can be considered more difficult than the other concepts. Consequently, most students will not satisfy the prerequisites for these concepts, and so the number of correctly diagnosed concepts decreases. However, no relation was found between the number of incorrectly diagnosed concepts and the number of prerequisite concepts. Therefore, the systems performance does not decline for the more difficult concepts.

Table 5-7: Number of prerequisite concepts of each category for each concept

Concept	Easy	Intermediate	Hard	Total
C1	0	0	0	0
C2	0	0	0	0
C3	0	0	0	0
C4	1	0	0	1
C5	1	0	0	1
C6	3	0	0	3
C7	3	2	0	5
C8	1	1	0	2
C9	1	1	0	2
C10	2	2	2	6
C11	1	1	1	3
C12	1	1	2	4

5.3 Evaluation with Real Students

5.3.1 ANOVA

ANOVA is the “Analysis of Variance,” which is used to determine whether in-group differences are significantly large after accounting for differences in the variances within groups. More precisely, ANOVA compares the differences among group means by decomposing the total variance in the data into within-group variance and between-group variance. If the between-group variance is sufficiently larger than the within-group variance, then ANOVA concludes that there are differences between the means of the groups.

Table 5-8: ANOVA Table Parameters

ANOVA Table				
Source of Variation	SS	DF	MS	F
Between	$SS_{between} = \sum_{j=1}^{j} n_j (\overline{Y_j} - \overline{Y})^2$	$j-1$	$MS_{between} = \frac{SS_{between}}{j-1}$	$\frac{MS_{between}}{MS_{within}}$
Within	$SS_{within} = \sum_{j=1}^{j} \sum_{i=1}^{n_j} (Y_{ij} - \overline{Y_j})^2$	$n-j$	$MS_{within} = \frac{SS_{within}}{n-j}$	
Total	$SS_{total} = \sum_{j=1}^{j} \sum_{i=1}^{n_j} (Y_{ij} - \overline{Y})^2$	$n-1$		

Of which, $SS_{between}$ measures variations of the group means around the overall mean (between groups), SS_{within} measures the variation of each observation around its group mean (Within groups) and SS_{total} measures variation of the data around the overall mean (total).

DF (Degrees of Freedom) is the factor that adjusts for how large the groups are and the number of groups being considered: *Number of groups* (j) -1 for $SS_{between}$, *Sample size* (n) - *Number of Groups* (j) for SS_{within} , and *Sample size* (n) -1 for SS_{total}.

MS (Mean Square) = SS/DF is the standard deviation. Its numerator is the sum of squared deviations (SS), divided by the appropriate number of degrees of freedom (DF).

F (F-Statistic or F-Ratio) = MSG/MSE, tells the proportion of variation between the groups compared to the variation within the groups. In general, the larger the F is, the more likely the variation between the groups is significant. The level of significance is determined by comparing it to the F-Critical value for the samples. If the F-Statistic is larger than F-Critical, then the variation between the groups is statistically significant.

5.3.2 Introduction about the Experiment

Evaluation is widely considered an important and challenging research issue in the area of IES in general. In fact, the lack of evaluation data, as well as the difficulty in generalization and the resulting difficulty in the reuse of successful design practices, constitutes one of the main barriers for IES to become mainstream technology. The evaluation of specific models and adaptive systems has been widely discussed [158, 159] and the traditional "with or without" approach is considered the gold standard, where experiments are conducted between two groups of users, one working with the adaptive application, the other with its non-adaptive version, assuming, of course, that an adaptive application can be easily decomposed into its adaptive and non-adaptive components.

This evaluation focuses on evaluating the effects of Bayesian network diagnosis and inference mechanisms by speculating methods to select learning materials for students. The experiment was conducted during the regular instruction of an engineering course: Integrated Digital Design. Fifty-eight of the third-year students from the ECE Department participated in the study. They were all taught by the same lecturer and were randomly assigned to use different tutoring systems to do their after-class tutorials. All participants took a pre-test before their evaluations. They were randomly divided into three categories: fully experimental group, partial experimental group and control group.

- *Control group (CG):* Sixteen students attended the conventional paper-and-pencil tutorial.
- *Partial experimental group (PEG):* Twenty-one students worked at their own pace using the web-based tutoring system with adaptive concept selection, but random question selection.
- *Fully experimental group (FEG):* Twenty-one students worked at their own pace using the web-based adaptive tutoring system with adaptive concept selection and adaptive question selection methods.

The experiment evaluated the effectiveness of the adaptive question selection method by comparing the results of FEG with PEG, and the effectiveness of adaptive concept selection method by comparing the results of PEG with those of CG. The overall tutoring system is evaluated by comparing the results of FEG with CG.

Table 5-9: Test Statistics of the three groups

Group	Pre-test score (full mark =100)		Post-test score (full mark =100)	
	Mean	Std	Mean	Std
FEG	70.4	0.102	85.8	0.077
PEG	73.4	0.141	78.5	0.121
CG	70.6	0.133	67.7	0.145

ANOVA was applied to make sure that the three groups had the equivalent initial competencies based on their pre-test scores. The mean and standard deviation of each group is listed in Table 5-9. ANOVA is an approach that allows a judgment if there is a significant difference between a number of groups. No reliable difference was found between the two groups with

$$F_{PEG,FEG}(1,40) = 0.637,\ p = 0.429,$$
$$F_{PEG,CG}(1,35) = 0.387,\ p = 0.538,$$
$$F_{FEG,CG}(1,35) = 0.002,\ p = 0.962.$$

In Table 5-10, the F statistic is 0.637, and this has a p-value of 0.429. Since this is larger than 0.05, we concluded that there was indeed no significant difference between the FEG and PEG groups of students. The ANONA analysis of reliable difference for PEG and CG is shown in Table 5-11, and the analysis of reliable difference for FEG and CG is shown in Table 5-12.

Table 5-10: ANONA analysis of FEG and PEG

SUMMARY						
Groups	Count	Sum	Average	Variance		
PEG	21	14.779	0.704	0.010		
CG	21	15.415	0.734	0.020		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.010	1	0.010	0.637	0.429	4.085
Within Groups	0.606	40	0.015			
Total	0.615	41				

Table 5-11: ANONA analysis of PEG and CG

SUMMARY						
Groups	Count	Sum	Average	Variance		
PEG	21	15.415	0.734	0.020		
CG	16	11.29	0.706	0.018		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.007	1	0.007	0.387	0.538	4.121
Within Groups	0.664	35	0.019			
Total	0.6719	36				

Table 5-12: ANONA analysis of PEG and CG

SUMMARY						
Groups	Count	Sum	Average	Variance		
PEG	21	14.779	0.704	0.010		
CG	16	11.29	0.706	0.018		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0．0003	1	0．0003	0.002	0.962	4.121
Within Groups	0.473	35	0.014			
Total		36				

5.3.3 Results Analysis

We designed pre- and post-tests to assess students' ability to solve tutorial questions. The data and results generated were analyzed in different ways.

1) Post-test relationships

Table 5-13: ANONA analysis of post-test relationships between FEG and PEG

SUMMARY						
Groups	Count	Sum	Average	Variance		
FEG	21	18.014	0.858	0.006		
PEG	21	16.5	0.786	0.015		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.055	1	0.055	5.227	0.029	4.085
Within Groups	0.417	40	0.010			
Total	0.472	41				

ANONA was applied to identify the post-test relationship between FEG and PEG $\alpha = 0.05$ to see if the Bayesian adaptive question selection method was more effective. Based on these results, the post-test results demonstrated a significant difference between the two groups $F_{FEG,PEG}(1,40) = 5.227, p = 0.029,$ as shown in Table 5-13, and the average percentage of FEG (0.858) was higher than that of PEG (0.786). The effectiveness of the adaptive concept selection method is identified by the ANOVA applied to the post-test results of PEG and CG with $F_{PEG,CG}(1,35) = 6.077, p = 0.018,$ as shown in Table 5-14, and the average percentage of PEG (0.786) was higher than that of CG (0.677). Therefore, we concluded that after learning with the Bayesian tutoring system, FEG performs better than PEG. Meanwhile, PEG showed a better performance than CG.

Table 5-14: ANONA analysis of post-test relationships between PEG and CG

SUMMARY						
Groups	Count	Sum	Average	Variance		
FEG	21	16.5	0.786	0.015		
PEG	16	10.84	0.677	0.021		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.107	1	0.107	6.077	0.018	4.1217
Within Groups	0.6147	35	0.018			
Total	0.727	36				

2) Learning gain

Students' learning gain is an important evaluation metric for an intelligent tutoring system. A common measure of learning gain with normalization is $\frac{post\ score - pre\ score}{1 - pre\ score}$ [134]. Pre- and post-tests are often used to measure learning gain. We use this method to evaluate the effectiveness of the various features of the Bayesian tutoring system.

The learning gains of the three groups were compared to evaluate the effectiveness of the adaptive feature of the Bayesian tutoring system. The mean of the FEG learning gain 0.437 with a standard deviation of 0.053 is higher than that of PEG with an average learning gain 0.117 and a standard deviation of 0.167. A reliable difference was found $F_{FEG,PEG}(1,40) = 9.819, p = 0.003,$ as shown in Table 5-15, and it indicates that there were statistically significantly differences in learning gains between these two different groups. In other words, after learning with adaptive tutoring methods, the learning gain of FEG performed better than PEG.

To evaluate the effectiveness of the adaptive concept selection method, the learning gain of PEG was compared to that of CG. The mean of CG was 0.205 with a standard deviation of 0.275. A significant difference was found $F_{PEG,CG}(1,35) = 4.435, p = 0.042,$ as shown in Table 5-16, between the learning gains of these two groups, PEG learning by the partially adaptive concept selection method performed better than CG, which learned by using the traditional tutorial method.

Table 5-15: ANONA analysis of learning gain between FEG and PEG

SUMMARY						
Groups	Count	Sum	Average	Variance		
FEG	21	9.175	0.437	0.053		
PEG	21	2.465	0.118	0.167		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	1.073	1	1.073	9.819	0.003	4.085
Within Groups	4.372	40	0.109			
Total	5.446	41				

Table 5-16: ANONA analysis of learning gain between PEG and CG

SUMMARY						
Groups	Count	Sum	Average	Variance		
FEG	21	2.465	0.117	0.167		
PEG	16	-3.283	-0.205	0.275		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.945	1	0.945	4.435	0.042	4.121
Within Groups	7.456	35	0.213			
Total	8.400	36				

3) Effect size

Effect size is a standard method of comparing the results of one pedagogical experiment with others [135]. The common method to calculate the effective size is to subtract the control group's mean gains score from the experimental group's mean gain score, divided by the pooled standard deviation for those means [136] as in the following equation.

$$\begin{aligned} \textit{Effect size} &= \frac{(\textit{mean of } x2) - (\textit{mean of } x1)}{\sigma_{pooled}} \\ &= \frac{(\textit{mean of } x2) - (\textit{mean of } x1)}{\sqrt{\left(\sigma^2_{x1} + \sigma^2_{x2}\right)/2}}. \end{aligned} \tag{5-1}$$

The mean and standard deviation values are listed in Table 5-9. The effect size of adaptive question selection compared to random question selection is $\frac{0.154 - 0.051}{\sqrt{\left(0.184^2 + 0.126^2\right)/2}} = \frac{0.103}{0.158} = 0.65$. The effect size of adaptive tutoring system compared to traditional tutoring method is $\frac{0.154 - (-0.029)}{\sqrt{(0.126^2 + 0.197^2)/2}} = \frac{0.183}{0.166} = 1.10$, which is comparable with the effect size of 0.63 for Conceptual Helper [137], 0.66 for SQL-Tutor [138] and 1.0 for Anderson's tutoring system [139]. However, it is still behind the human tutoring, which scored 2.0 [140].

Researchers have suggested a criterion to judge and conceptualize the effect size. For Cohen's effect size, 0.2 is a small, 0.5 implies medium and 0.8 and above indicates a large effect size [136]. So in our results, the effect size of 1.10 is acceptable and belongs to a significant effect size compared to the traditional classroom teaching method and signifies the effectiveness of the combinations of the concept selection, adaptive question selection and the adaptive feedback functions.

5.4 Survey Results

In addition to students' academic performances, students' feedback on the GWATS is another very important source of information to evaluate whether GWATS achieves its goals of promoting active learning in large classes. After working with the system, all students answered a questionnaire about the system and its effectiveness.

Table 5-17 shows that the usefulness of ATS is proven as 80% of the students agreed that ATS helped them understand and learn the module effectively. As for the effectiveness of ATS, 74.29% of students agreed that ATS improved their learning performance, 67.14% regarded ATS as a good supplement to classroom-based tutorials and 62.5% enjoyed learning with the system and suggested its use in other modules. Considering the facts that the system is only in a prototype stage and this is the first time that students were exposed to such a system without any training, these results are very encouraging. As we can see from their answers, students in the two different groups have distinct views of the system, especially on the personalized features like misconception identification, question adaptation, the usefulness of hints presented, etc.

Table 5-17: Feedback Analysis (in percentages)

Q1: Does the ATS help you understand and learn the module effectively?

Q1 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
All groups (%)	25.71	54.29	11.43	7.14	1.43

Q2: Do you agree that the ATS improves your learning performance?

Q2 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
All groups (%)	14.29	60.00	21.43	2.86	1.43

Q3: Do you agree that the ATS is good complements to classroom-based tutorials?

Q3 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
All groups (%)	20.00	47.14	22.86	7.14	2.86

Q4: Do you agree that the ATS should be used for other modules?

Q4 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
All groups (%)	12.5	50	26.39	5.56	4.17

Q5: Do you agree that the ATS is a better way to diagnose the misconception you might have?

Q5 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
PEG (%)	0	18.18	50.00	27.27	4.55
FEG (%)	12.00	66.00	22.00	0	0

Q6: Do you agree that the ATS is able to present questions according to your ability?

Q6 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
PEG (%)	0	32.00	28.00	32.00	8
FEG (%)	7.84	78.43	13.73	0	0

Q7: Do you agree that the hints provided by the system help you solve the problem?

Q7 Answer	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
PEG (%)	16.00	36.00	32.00	8.00	8.00
FEG (%)	34.00	40.00	22.00	4.00	0

The great majority of the surveyed students in FEG agreed with these personalized characteristics, while around half of students in PEG disagreed. As for misconception identification capability, 78% in FEG agreed that ATS can diagnose their misconception, and in this group, nobody objected to this. In PEG, only 18.18% agreed with this. In FEG, 86.27% of the students surveyed agreed that the questions presented by the system conformed to their abilities, while only 32% students from PEG agreed and 40% disagreed.

The significant differences in the opinions from the two groups are due to two distinct features of ATS. First, the adaptive question selection method used in FEG is capable of presenting questions according to the individual's capabilities and knowledge states. These adaptive questions can better diagnose the mastery states of the student. Second, the adaptive presentation and feedback tutoring model interprets the student's needs and possible misconceptions and generates relevant information to meet the individual's needs. This feedback clarifies the student's strengths and weakness and provides a corresponding method to help them overcome their weakness, which naturally promotes learning. However, for students in PEG, questions were randomly selected without taking the individual's knowledge states into account. Hence, they could not experience these two adaptive features.

One interesting result is that 74% from FEG thought that the hints presented were helpful in solving problems, and more than half of PEG agreed with this, which illustrates the usefulness of the hints. The hints presented to students were in different levels from generic to specific, and the usefulness of multiple hints was sensed by all students.

5.5 Conclusion

The evaluation of the ATS suggests that the adaptive tutoring strategies followed by the underlined generic tutoring model, including concept selection, prerequisite relations filtering and question selection, are effective in diagnosing the student's knowledge states, adaptively selecting and delivering learning materials to students and providing feedback. The effectiveness of the adaptive question selection method was evaluated by ANOVA, which showed a significant improvement on post-test results and the learning gains of the fully experimental group (FEG) and the partial experimental group (PEG). The ANOVA applied on the significant improvement on post-test results and learning gains of PEG and the control group (CG) showed the value of other adaptive features of the adaptive tutoring system. Effect size of the adaptive tutoring system is 1.10, which is quite acceptable. Survey results showed the satisfaction of students learning with the system, approval of its effectiveness and favorable acceptance of the system from the students' point of view.

GWATS is continuously being developed and improved. Its effectiveness will be further evaluated by real students for other modules in future.

CHAPTER 6

PROTOTYPE OF MOTIVATIONAL TUTORING SYSTEM

It is well known that students with high intrinsic motivation often outperform those at low motivation state. Therefore, students should be kept motivated in order to achieve optimal learning results. Learning efficiency and engagement are highly correlated with and influenced by students' temporal cognitive characteristics such as emotion, interest, attitude, and learning style, etc. Psychological characteristics, like emotion and motivation play important roles in the learning process as well. Experienced teachers are capable of observing students' learning behavior, inferring their motivational states and then react accordingly to keep them motivated. Computer-based online learning system is not born with this ability but endowed with monitoring and memory skills. With these skills, students' motivation can be assessed through the observation of the interactions between the student and the education system [80] [140] [141]. Therefore, GWATS can take students' motivation states into account and work more intelligently.

The GWATS architecture described in Chapter 3 including behavior tracking and analysis modules, is trying to derive the student's motivational states from the learning behavior and take the derived motivational states into account to provide better adaptive tutoring. The implementation of behavior tracking module and the detail are presented in section 3.5. Behavior analysis is a very complicated process, which needs the support of artificial intelligence or data mining techniques. In this chapter, we present a prototype of motivational tutoring system based on GWATS architecture to demonstrate the effectiveness of the behavior analysis module and the better performance obtained with the prototype system

6.1 Description of the Prototype System

The prototype system is built based on the GWATS architecture and the cooperation of all the comprising modules. Learning session starts with the student choosing one target concept as learning objective and ends with the student submits tutorial answers. Behavior tracking module tracks key actions initiated by the student and important parameters during each session, such as time taken on each tutorial question, hints checked, and cases where he/she gives up etc, and then stores in the database. After current session is over, behavior analysis module retrieves student's learning behaviors occurring in the latest session and infers the student's confidence, independence and effort based on heuristic rules. Then the values of these variables are fed to the tutoring model to make pedagogical decisions not only based on the student's knowledge-related states but also considering the motivational states.

Learning behaviors tracked actually happened in the previous session. Without any doubt the values reflect the states of the previous session. How to infer current motivation from the previous motivation states and make the most appropriate pedagogical decisions adaptive to the student's current states is our mission. Since student's knowledge mastery states and motivation states evolve over time, we can use the Dynamic Decision Network (DDN) to model these changes and to infer the current motivational states from the previous states. The basic knowledge about DDN and how to infer motivational states from learning behaviors will be described in next sections.

6.2 Infer Motivational States from Learning Behaviors

Del Soldato and Du Boulay's motivational planning approach [142] is widely explored in learning environment. It correlates a motivational planner with the domain-based instructional planner to make pedagogical decision based on both domain-related and motivation-related states. This approach models three motivational variables: effort, confidence and independence. It develops production rules to model the three motivation variables based on student's learning behaviors in the learning environment. In GWATS, student's learning behaviors tracked can be used to infer student's confidence, independence and effort based on motivational planning approach.

In current GWATS, during each session, the student could read the definition of the target concept, view the example or exercise, go through the multimedia simulation/visualization, do the virtual experiment or attempt the tutorial questions to assess his/her mastery states of the target concept. After the student clicks the "Attempt Tutorial questions" button, GWATS will automatically select and present a series of appropriate questions for students to attempt. Student can answer the question directly if he/she acquires sufficient knowledge related to the question. Or he/she can request to view the hint of the target concept or the hint to each question by clicking the "hint" icon, if he/she is not so confident about his/her answer. Hint of the concept provides general information of the concept, while hint of the question provides specific information to the specific question. Requesting hints of different level can tell the level of help needed by the student. After the student submits the answers, he/she will receive an instant feedback including the updated mastery states of the target concept and other related concept(s), the correctness/incorrectness of the answer and solution to each question. Student can attempt another tutorial at the same level if he/she hasn't mastered the target concept, or move to the un-mastered related concept, or view the solutions. Student can also view his/her past tutorial histories. Student's actions are recorded in the server database, including all the links or buttons clicks, sequences, and the intervals between clicks. During the interaction the following observable attributes are logged:

Student learning activity: such as, student reading the definition, viewing the examples, doing the exercises, or viewing the simulation/visualization, which indicates that the student is engaged in actions aimed at understanding and mastering the target concept.

Time taken: time spent on each learning action, which is widely used as general indicator of the engagement.

Help request: student clicking on "hint", which is indicator of the student's effort or confidence. This behavior falls into different categories according to the timing the help is request: help before attempting, help after choosing the wrong answer and help after keying in the correct answer. Hint before attempting might indicate that the student is not so confident. But requesting hint after choosing the answer might indicate that the student does put some effort and want to understand and solve the question.

Giving up: student gives up a question before attempting to solve it. If the question is within his/her ability according to his/her knowledge model, this action might show that the student does not engage in and just wants to gets over the learning session. If the question is beyond his/her ability and perceives as difficult, this might suggest that he/she doesn't believe his capability of solving the difficult question.

Guessing: the student selects answers within very short time (basically 10 seconds), no hint requested. This might show that the student doesn't think the question carefully and just randomly selects an answer until the correct answer is shown.

Viewing solution: the student clicks to see the solution to each question after submitting the answers. This might imply that the student shows serious attitude at learning and is anxious to know the solution.

6.3 Motivation States Modeling

Student's motivation states: effort, independence and confidence can be detected because these internal states usually affect learning behaviors. For example, the student with extremely high confidence at the moment believes that he/she can successfully solve the tutorial questions by him/herself, thus there will be no help request during the interaction. But for those who just experienced failure and do not have any confidence at all, he/she will ask for help before attempting to solve a problem. Hence we can infer effort, confidence and independence from student's learning behaviors initiated in the education system. Motivation states actually are complex to understand and difficult to detect, here we are not aiming at detect all the motivation states exactly but presenting a very simple prototype system – Motivation based Adaptive Tutoring System (MATS), which is trying to detect the motivation states and respond accordingly to see whether considering the motivation states will improve the student's learning performance.

In MATS, only confidence, independence and effort will be modeled and detected. We employ rules in Del Soldato and Du Boulay' motivational planning approach to derive student's motivation states based on fuzzy rules. MATS is based on assumption that students have general response patterns under the similar circumstance to simply the construction and deduction.

6.3.1 Modeling Confidence

Student's confidence is determined mainly based on the tutorial solving behaviors, like knowledge performance, given-up and help requested. According to rules in [142], confidence is represented as a value (conf-value) in a linear scale, and the limits for the lowest and the highest possible confidence values are predefined by the instructors. In MATS, each tutorial might include more than one question with different difficulty levels. Tutorial marks don't accurately reflect the knowledge state of a student. But the mastery state of the target concept which takes the question difficulty level into account is the best parameter to represent the knowledge performance. The student's initial confidence value is set as 5 which is the medium level. Confidence is incremented or decremented according to the modeling rules. Rules to modeling confidence are as followings:

> *If the student mastered the target concept without help then the student's confidence will be increased by 2.*
>
> *If the student mastered the target concept with help then the student's confidence will be increased by 1.5.*
>
> *If the student partially mastered the target concept without help then the student's confidence will be increased by 1.*
>
> *If the student partially mastered the target concept with help then the student's confidence will be increased by 0.5.*
>
> *If the student did not master the target concept with help then the student's confidence will be decreased by 1.5.*
>
> *If the student did not master the target concept without help then the student's confidence will be decreased by 2.*

If the student gave up one difficult question then confidence will be decreased by 0.5.

If the student gave up one medium level question then confidence will be decreased by 1.

If the student gave up one low level question then confidence will be decreased by 0.5.

6.3.2 Modeling Effort

By estimating how much time the student has already spent on a task, a human tutor can infer the student's effort put on his task. Imitating how the human tutor infers the student's effort during the in-person interactions, we derive the inference rules to detect the student's effort. Variables used to model student's effort are: average time-taken, average hint-requested, average given-up and the mastery states of the target concept.

If gave-up without try a question but time-taken is less than expected-time then effort will not change;

If gave-up without try a question but time-taken is greater than expected-time then effort will be increased by 0.5;

If target concept is partial-mastered or mastered and time-taken is less than expected- time then effort will be increased by 1;

If target concept is partial-mastered or mastered and time-taken is greater than expected- time then effort will be increased by 1.5;

If the student requested a help then effort will be increased by 1;

6.3.3 Modeling Independence

Variable used to model student's independence is hint-requested. Modeling rules as following:

If the student requested a generic level help then independence will be decreased by 0.5;

> *If the student requested a specific level help then independence will be decreased by 1.*

After each session is over, the mastery state of the target concept is inferred based on the tutorial results. All the other variables, such as time-spent, hint-request and given-up, needed to infer confidence, effort and independence can be derived from the student's learning behavior tracked by the Behavior Tracking model. Values for confidence, effort and independence are updated based on the above modeling rules. The responding states, like high, medium or low, can be determined based on the fuzzy rules.

The motivation states are not static, and they may change over time as a result of the changing circumstances. For example, the student may feel more confident when he gets good performance and may not feel so confident after make several mistakes. Since motivation states remain relatively stable in certain period of time, the influence of previous states should be considered when modeling student motivation states. That's why we use the dynamic Bayesian network to model the dynamic nature of the motivation states and its affection on the next state. In our dynamic Bayesian network we maintain two time slices. The motivation states of time slice t_{n-1} are inferred based on the student's learning behavior at that time and then used to predict the motivations at time t_n.

For the sake of simplicity, the model we presented before includes only a subset of the variables taken into account to assess the motivation reactions in Del Soldato and Du Boulay' motivational planning approach [142]. We use heuristic rules to model the relationships between motivation factors and the related variables. The objective is to give an intuition of how the motivations are inferred and how to be used in adaptive tutoring. With more information collected and available, more and more variables should be introduced to make the model accurate and comprehensive.

6.4 Implementation of the Prototype System with DBN

A BN is useful for problem domains where the state of the nodes is static. In such domain, nodes represent random variables and edges represent conditional dependencies. Each node stores a conditional probability table giving its node parents. But the dependencies and the conditional probabilities in BN are just for a particular point in time and BN does not provide direct mechanism for representing temporal dependencies. However, student's motivational states are dynamically changed and evolving over time, BN seems not sufficient. **Dynamic Bayesian network** (DBN), which is a BN that represents sequences of variables with a time dimension, can be used to model dynamic systems [143]. In this section, we only describe the DBN formalism that is in common use today. A more extensive introduction on DBNs can be found in: [144] [145].

6.4.1 Dynamic Bayesian Network

The dynamic extension does not mean that the network structure or parameters changes dynamically, but that a dynamic system is modeled. A DBN is a directed, acyclic graphical model of a stochastic process. It consists of time-slices, with each time-slice containing its own variables. Each time slice of a DBN corresponds to one particular state of the system and the movement between the slices reflects a change in state evolving with time. A DBN can be defined as the pair ($B1, B \rightarrow$) where $B1$ is a BN that defines the prior or initial state distribution of the state variables $p(Z_1)$. Typically, $Z_t = (U_t, X_t, Y_t)$ represents the input, hidden and output variables of the model. $B \rightarrow$ is a two slices temporal Bayesian network (2TBN) that defines the transition model $p(Z_t \mid Z_{t-1})$ as follows:

$$p(Z_t \mid Z_{t-1}) = \prod_{t=1}^{N} p(Z_t^l \mid Pa(Z_t^l)), \tag{6-1}$$

where Z_t^i is the i-th node at time t and could be a component of X_t, Y_t or U_t. $Pa(Z_t^i)$ are the parents of Z_t^i, which can be in the same or the previous time-slice. The nodes in the first slice of the 2TBN network do not have parameters associated with them. The nodes in the second slice do have a CPT. The structure repeats and the process is stationary, so the parameters for the slices t = 2, 3 . . . remain the same. This means that the model can be fully described by only giving the first two slices. In this way, an unbounded sequence length can be modeled using a finite number of parameters. The joint probability distribution for a sequence of length T can be obtained by unrolling the 2TBN network:

$$p(Z_{1:T}) = \prod_{t=1}^{T} \prod_{t=1}^{N} p(Z_t^i \mid pa(Z_t^i)) \tag{6-2}$$

In Netica, the user designs the DBN and then compiles it, after that, inference is possible. Compiling the DBN basically means that it is converted to a junction tree representation. When designing a DBN, temporal arcs can be added between nodes. After defining the DBN, it can be unrolled for t slices and compiled. The resulting BN is opened in a new window. Inference can be performed on this unrolled and compiled BN. We can enter evidence by hand or by importing a data file.

6.4.2 Modeling Motivation States using DBN

In our model, in order to keep the size of the network under control, only two time slices are maintained at a given time by applying the rollup mechanism, because the influence of previous time slices can be summarized as prior probabilities in the first of the two active time slices. The link between emotion nodes across time slices models the evolution of emotions, representing the fact that a previous emotion affects a student's subsequent emotion. Every time we apply rollup mechanism by setting the prior probabilities of the root nodes at time slice t_{i+1} as the posterior probabilities of the corresponding root nodes at time slice t_i.

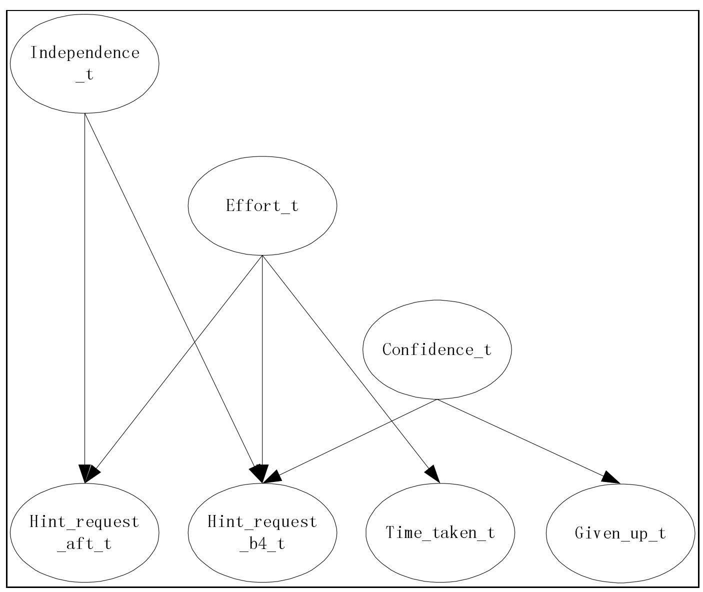


Figure 6-1: **DBN for MATS tutoring model**

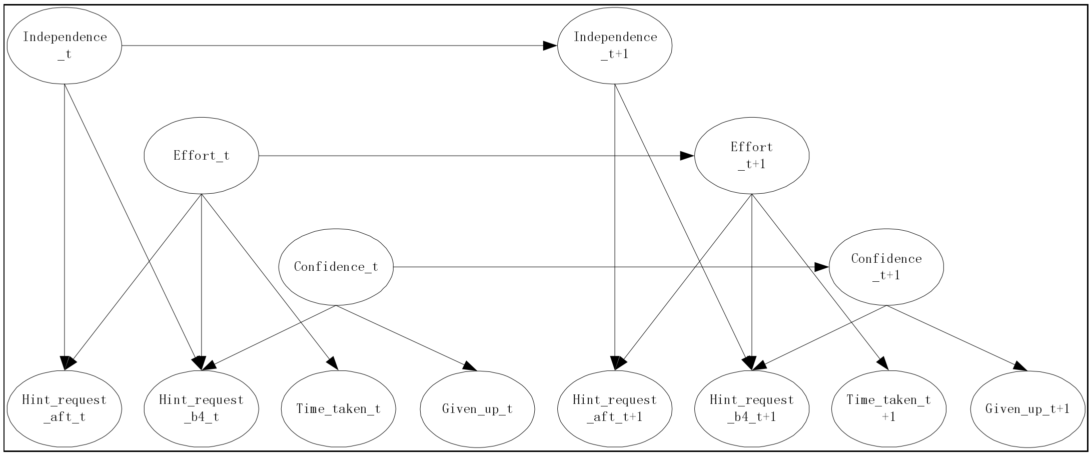


Figure 6-2: **The 2TBN for MATS tutoring model**

Student's learning behaviors happening in current session can be used to infer his/her motivation states for the moment, but these behaviors can't be captured until the session ends. If we consider a student's motivation states relatively stable, we can infer his/her motivation states before next session from the states of previous session. Therefore there are two steps to deduce student's motivation states for the coming session. First, inferring his/her previous states based on learning behaviors tracked in previous session; then modeling current states evolving from the previous session. Take the $(T+1)^{th}$ learning session for example, the initial state of confidence variable for $(T+1)$ slice can be inferred from the updated probability of the confidence after the T^{th} session ending based on the transitional probabilities. The initial state will be affected by the learning variables happened during the T^{th} session and be updated based on the learning behaviors to deduce the confidence variable after $(T+1)^{th}$ session, which will be used as the initial state for $(T+2)^{th}$ session.

6.5 Making Pedagogical Decision with DDN

As mentioned in the introduction part, lots of research energy about motivational factors has been put on how to detect student's motivations, but less explicitly dealing with how to use the detected motivational states to promote personalized tutoring or individual learning. We will present the working mechanism of making pedagogical decisions based on both the knowledge states and the student's motivational states in this section. To do this, we put the knowledge state as chance node in the DBN and extend DBN with two additional types of nodes: utility and decision nodes to form Dynamic Decision Networks (DDNs). DDNs model decisions for multiple situations in which decisions, attributes or preferences can change over time. Decision node contains possible alternatives standing for the actions that the decision can take to achieve the desired outcome. Utility node contains a function to calculate utility based on variables and decisions, and holds a table of utility values for all configurations of its parent nodes. These networks have been used in a variety of applications including clinic decision making [146] and intelligent tutoring systems [147] and planning [148], etc.

The process of using DDN to make decision is as follows [142]:

1. Set the evidence variables for the current state.
2. For each possible value of the decision node, set the decision node to that value.
3. Calculate the posterior probabilities for the parent nodes of the utility node using a standard probabilistic inference algorithm and
4. Calculate the resulting utility function for the action.
5. Return the action with the highest expected utility as the most appropriate action for the given situation.

6.5.1 DDN for Prototype System

For sake of simplicity, we build a DDN to implement a very simple tutoring task as a subsection of tutoring model to show the working mechanism of the DDN and the performance of considering motivational states. We have identified 16 variables interested in the two-slice DDN, including two decision nodes and two utility nodes, as shown in Figure 6-3.

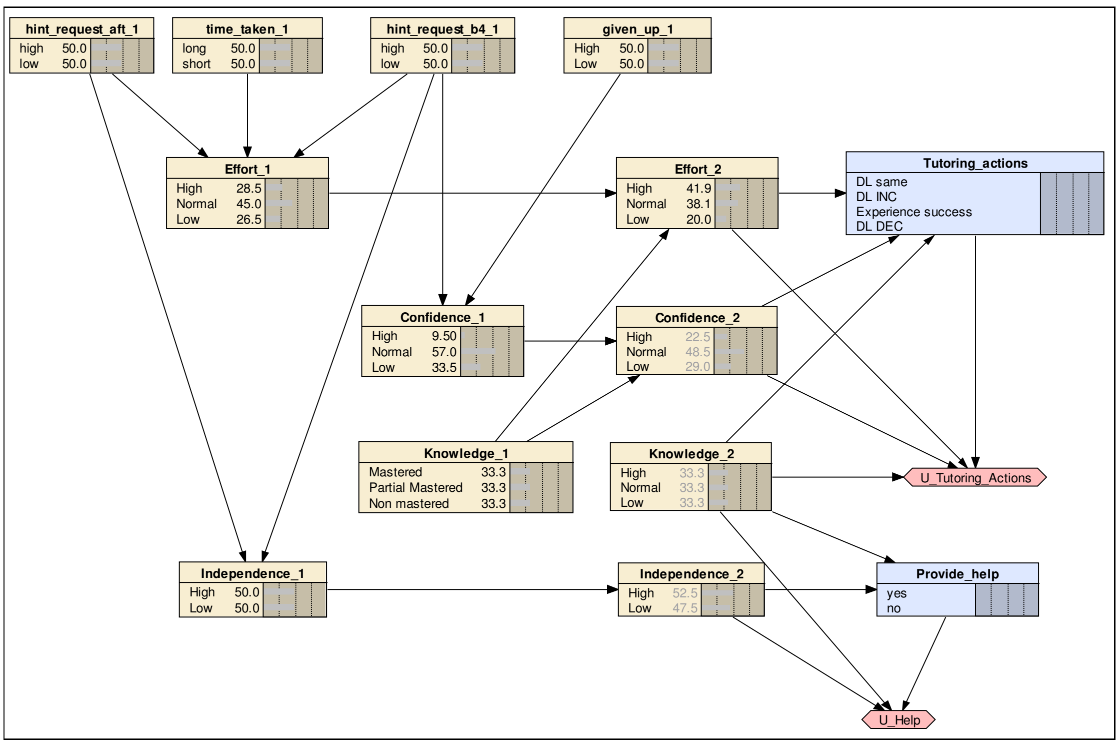


Figure 6-3: **The DDN for MATS tutoring model**

This DDN is aimed to choose the most appropriate difficult level of tutoring questions for the student, given his current knowledge states and motivational states. Knowledge node is included in the DDN representing the student's current mastery states of the target concept. If the student successfully mastered the target concept C_t at time t, the tutoring model will select another concept C_{t+1} different from the previous one as the target concept for slice *t+1*. Since C_t and C_{t+1} might be different, the knowledge states in the two slices might be different as well. No matter whether they are different or not, the probability of the knowledge state node can be simply copied from the static Bayesian network. As for the transitional probability, if C_{t+1} is the same as C_t, the corresponding transitional probabilities should be set to one, since there is no change at all during the two sessions. But if C_{t+1} is different from C_t, corresponding transitional probabilities are the conditional probabilities of the C_{t+1} given C_t..

Tutoring_actions is a decision node representing the tutoring action on the difficulty levels (DL) of the tutoring questions selected for the next session. The action items are: “keep the same DL”, “DL increase”, “DL decrease”, and “Experience success” which means that the tutoring system should select easy questions in order to boost student’s confidence through good experience. Providing_help is another decision node which decides whether to provide help to student or not. If the student requests help too often in the previous or current session, the decision node will not provide help any more to encourage him/her to attempt the tutorial questions all by him/herself.

Utility node - U_Tutoring_Actions represents tutor preferences regarding DLs of the tutoring questions. Total utility is a weighted sum of the utilities for each affecting state component: effort state, confidence state and knowledge state, for the given situation. The utility values can be obtained by consulting experts or from the preference of decision making. In our prototype system, lecturers can manually assign the utility values with their preferences values based on their expertise to reflect different teaching strategies preferences. Since those three variables affecting the tutoring action decision node, the decision node will take all the three utilities values into account and will choose the one with maximize utility values as the final outcome decision. For example, when the student's confidence is diagnosed as being low but his knowledge state is assessed as high, and if the lecturer takes confidence as the most important aspect, the main goal will be to help the student regain a reasonable confidence instead of promoting difficulty levels of tutoring questions to challenge the student, which is the typical case in the traditional tutoring system that do not take motivations into account.

6.5.2 Conditional Probability Table Creation

For our prototype implementations, we used our best judgment to set initial values for CPT parameters, prior probabilities and utilities, leaving obtaining more accurate values as an important goal for future research. The action selection engines also accept an optional file to specify any probability or utility values that differ from the defaults.

For Effort, Confidence and Independence nodes, CPTs were constructed following the heuristic rules in Del Soldato and Du Boulay' motivational planning approach [13]. The parameters of our DBN include the prior probabilities for the student's motivational nodes, the conditional probabilities for the links within the same time slice, and the transitional probabilities of the motivations states between the two consecutive time slices. Since our study focuses more on the working mechanism rather than the accurate model, most of the required probabilities are falling back on expert opinion, either specified manually or deduced from heuristic rules. In particular, the prior probabilities for all three motivational state nodes are set to (0.5, 0.5). The transitional probabilities are specified accordingly considering that the student's motivational states remain relatively stable. For example, the transitional probability between the same states of two slices, e.g., high to high for confidence or independence, is high. However, the transitional probability between opposite states, high to low or low to high, is much lower correspondingly. Other conditional probabilities are determined subjectively in a similar mode.

The model described in this model covers in detail only slice t_{i+1} of the general model, and includes only a subset of the variables that are necessary to completely specify this time slice. We chose this subset to illustrate how the model is built and of its working mechanisms. But several additional variables should be included to accurately model a real interaction. The conditional probability tables (CPTs) in the initial model were defined using our estimates. In the future, we need to find a way to refine them based on the results of prototype system study.

6.6 Evaluation

There are multiple factors that affect the difficulty level of tutoring questions presented to the student. To improve the decision making process in MATS, at least seem to be better. We present several cases to compare the different decisions made by MATS and the traditional ATS.

Case One in Figure 6-4 shows a student who has shown ability to master the previous target concept since the knowledge state is high, but he put “normal” effort, requested too many hints and gave up too often in the previous session, therefore the “effort” diagnosed as normal and confidence as low. For the traditional ATS, since the student seems to master the current concept very well, therefore, in the next session the tutoring model will present more difficult student to challenge the student. But for MATS, the knowledge state is just one of the factors it should consider. It will take into account the student’s motivations into account and aim to promote the student’s learning as well as keep the student motivated. Therefore, the final decision MATS making is to present tutoring questions the student answered correctly before and let him/her experience the success to increase his/her confidence, which we can see from the DDN shown in the following figure.

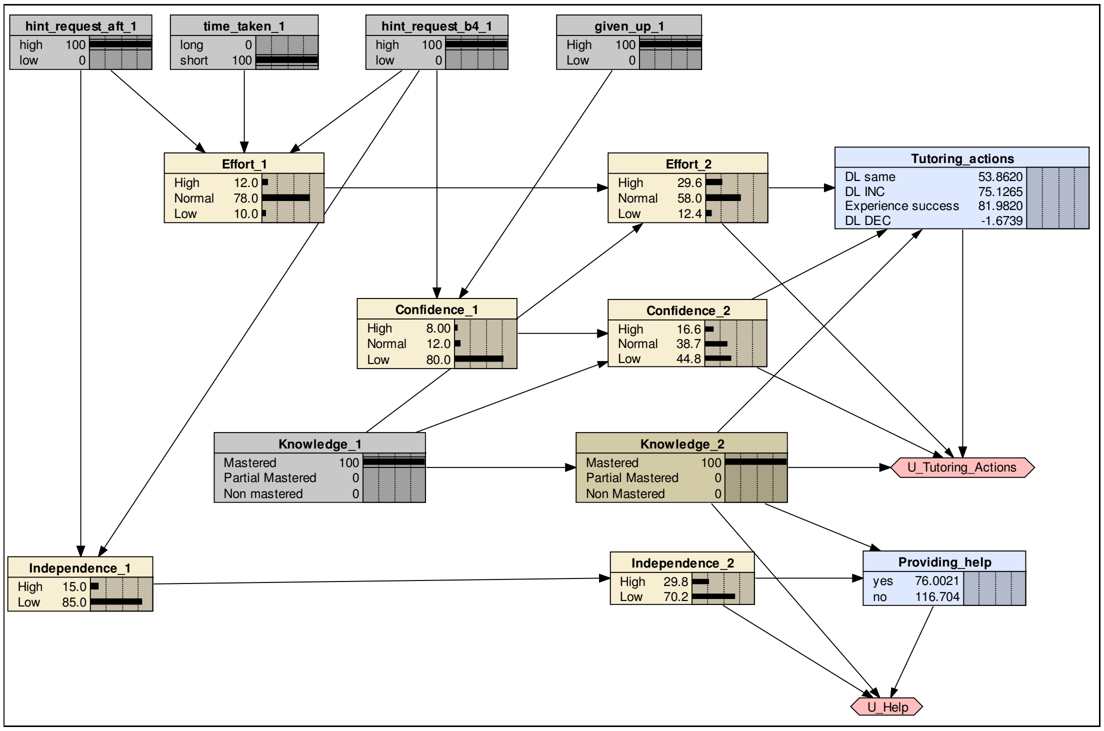


Figure 6-4: **The DDN for learning case One**

Case Two in Figure 6-5 shows that the student rarely requested hint, took short time to finish the previous leaning session and barely gave up, who is diagnosed as with high confidence and putting low effort. Since his knowledge state is diagnosed as partial mastery, for the traditional ATS, the tutoring model will judge that there is still space for the student to improve and will present tutoring questions with the same difficulty level for him/her to learn and review the previous target concept. But for the MATS, the low effort put and the partial mastered knowledge state might tell that the student felt boring with the easy questions. Therefore MATS will increase the difficulty levels and inspire the student to put more effort.

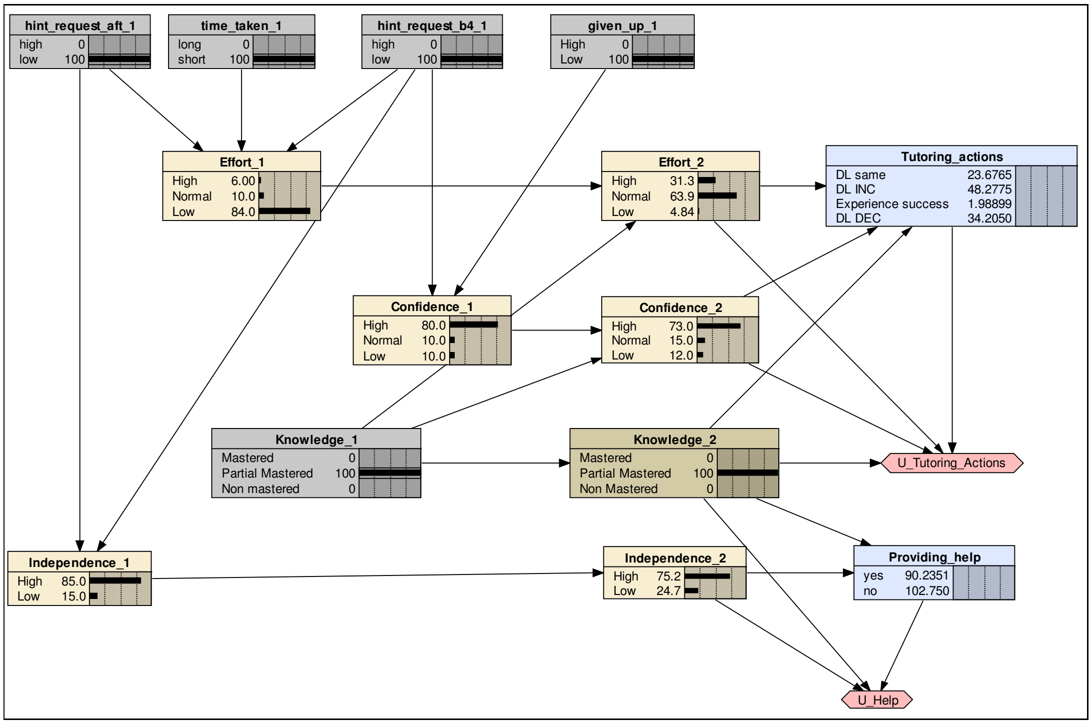


Figure 6-5: **The DDN for learning case Two**

Case Three in Figure 6-6 shows that the student requested hint a lot after attempting the tutorial questions, took long time to learn during the previous learning session, seldom requested hint before attempting tutorial questions and barely gave up, who is diagnosed with high confidence and has tried his best by MATS. Since the student did not mastered previous target concept, the traditional ATS will figure that questions presented to the student before were too difficult and beyond his/her capability and then will lower difficulty levels. MATS will consist with the traditional ATS and make the same decision by lowering the difficulty level of tutorial questions, since the student is highly motivated in high confidence and effort.

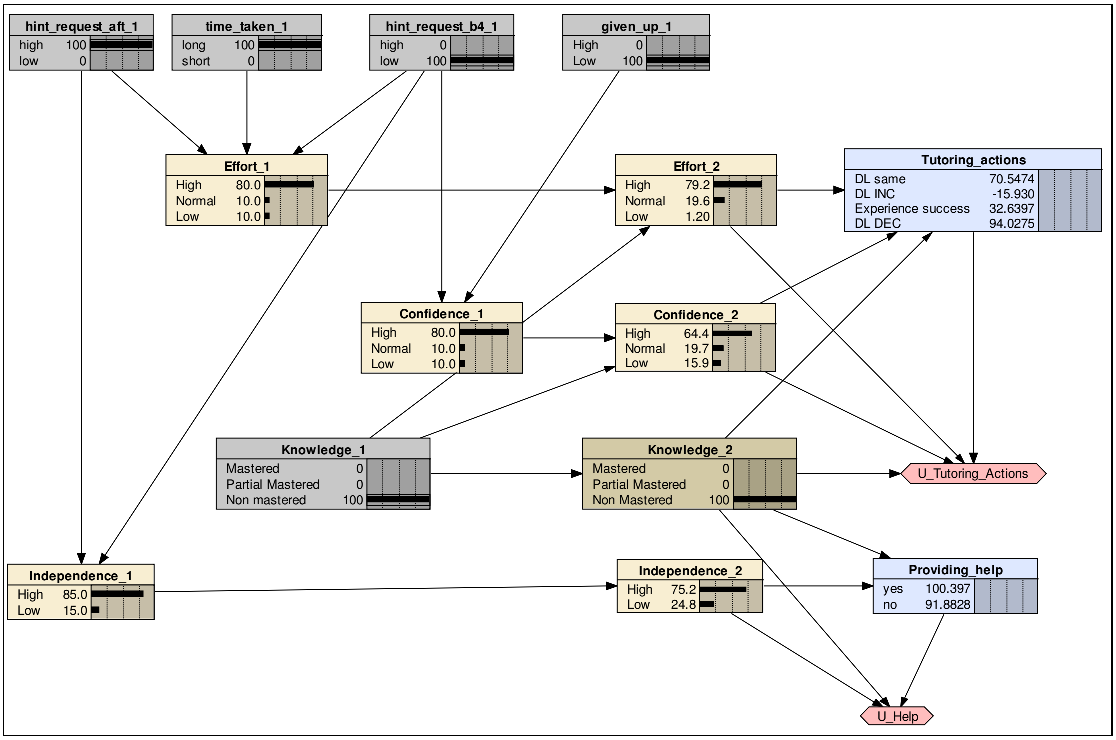


Figure 6-6: The DDN for learning case Three

6.7 Final Considerations

The performance of MATS and the efficiency of motivational states cannot be evaluated until it is put into real application. But MATS might make decisions different from traditional ATS given the same situation, because MATS consider the student's motivational states in addition to the knowledge state. Cases presented in previous subsection tell that MATS works at least as better as ATS.

There is still much work to do in order to achieve a strong student model and we are still far from that. However, the architecture of GWATS which has behavior tracking model and analysis model, and the previous results obtained using the prototype system give us strong confidence in motivational tutoring system might be a better system than the traditional one. We do believe that behavior analysis and motivational tutoring are not simple.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This chapter reflects on the research described in this thesis describes possible directions for research work.

The first topic of this thesis is the system architecture and the basic components of GWATS. The hierarchical structure of the domain knowledge, the modeling mechanism of the Bayesian student model, the underlying generic tutoring model and the provision and inclusion of the authoring environment into the infrastructure make it possible to conveniently construct an effective adaptive tutoring system for many subjects. Integration of the authoring environment into traditional adaptive tutoring system architecture is the first contribution of GWATS. With an authoring environment, an ATS can be modified whenever there is a need during the whole life cycle of the ATS. The authoring environment and its web-based characteristics not only simplify the ATS construction, but make the generated ATS easier to deliver and convenient to use.

With the layout built, the second aim is to show how to build an ATS based on GWATS. Procedures and underlying mechanism were presented. Currently, the authoring environment enables lecturers to create adaptive tutoring systems dedicated to certain groups of students for certain subjects by loading subject-related knowledge and students' characteristics information into the infrastructure to generate domain and student models. The other contribution of GWATS is its capability of compiling the constructed domain structure into a Bayesian student model. The separation of the domain structure with the knowledge content, the Bayesian overlay student model and the comparability of the ontology of the student model and the domain model remarkably simplify the student model construction procedure. Initially the compiling process was based on the heuristic rules. With the generated adaptive tutoring system put into application and the collected learning cases, it can learn more accurate learning parameters and the Bayesian student model.

Creating an ATS with GWATS can show its efficiency and usability but not its effectiveness. The ratio of the effort needed to create instructions to the sum of durations of all online instructions generated is 8.62:1, which is quite favorable and acceptable compared with 300:1 for the traditional CAI. This cannot tell the effectiveness but the efficiency of WAE, which is not enough. To verify and evaluate the effectiveness of the ATS is the third task of this research. Evaluations carried out to verify the effectiveness of the ATSs authored by GWATS constituted the second main part of this thesis. To achieve this objective, we presented the procedures of designing an instance ATS, applied it in the module Digital IC design language: Verilog. A number of qualitative and formative evaluations were performed to ensure that the ATS was usable, friendly to use and effective. Simulated students were employed to show the effectiveness of the genetic tutoring model and the result suggests that the adaptive tutoring strategies followed by the embedded generic tutoring model, including concept selection, prerequisite relations filtering and questions selection, are effective in diagnosing the student's knowledge states, adaptively delivering learning materials to students and providing feedback. Real students were surveyed, and the results showed that the adaptive tutoring system created by GWATS can be educationally effective in comparison with traditional instructional methods by improving the achievement levels. There is a significant improvement on post-test results and learning gains of the group that learned with the GWATS compared with the group using the traditional system, which shows the effectiveness of the adaptive tutoring system. Effect size of the adaptive tutoring system is 1.10, which is quite acceptable. Different opinions from the two different groups on the adaptive features verified the value and functions of the generic tutoring model and personalized characteristics. Survey results about the GWATS performance suggested its friendly usability and favorable acceptance. Evaluations showed that ATS authored by GWATS provides personalized tutoring resembling the process of one-to-one instruction in small classrooms for each individual, including intelligently identifying students' learning status, adjusting teaching strategies and learning contents to suit individual learning styles. In short, the ATS can promote learning performance in large classes.

The architecture of GWATS provides behavior tracking and analysis modules for capturing students' dynamic learning characteristics and states. The final part of this thesis presents a prototype system, MATS, built on the basic architecture of GWATS. But the MATS implements the behavior analysis module using a dynamic Bayesian network, diagnosing motivational states based on the learning behavior tracked and using a dynamic decision network to take students' knowledge and derived motivational states into account to provide more adaptive tutoring and keep students motivated during the learning process. This prototype system focused on how to take derived motivational states into consideration to make better decisions, instead of how to derive the motivational states from learning behaviors. Therefore, the parameters of DBN and DDN fall back on experts' opinion and there is still lot of work to do and plenty of room for improvement. The prototype system is not yet implemented, and it is not possible to make a more complete validation of the effectiveness of motivational tutoring. The three learning cases can only describe the working mechanism of MATS. MATS needs to be put into real use to see if there is a significant improvement in students' overall learning performance.

7.2 Future Work

The usability, efficiency and effectiveness of the WAE have been presented in this thesis. The future work of the WAE is to learn the CPT of the generic Bayesian student model from collected learning cases and to evaluate whether the revised Bayesian student model will get better results than the original one, the parameters of which are determined based on the prerequisite relationships and weights keyed in by the lecturers. As for the dynamic student model, we should learn CPTs from the tutorial answers submitted and students' learning states to revise the CPTs calculated by the parameters keyed in by the lecturers.

With WAE, GWATS seemed to be appropriate for many different domains. However, it is necessary to study how it should be adapted to different domains. For this stage, we believe that GWATS could serve as a starting point, but refinement is needed to adapt it to a particular domain. Its generic nature has yet to be validated in real teaching applications.

WAE considers individual learning styles and provides adaptive tutoring by presenting an interactive and hands-on learning environment to facilitate the individual's learning style. In the future, some work can be done to derive a student's learning style by identifying the learning patterns based on the collected learning behaviors.

Alternatively, we can use sequential pattern mining and user clustering methods to classify students into different groups and then use case-based reasoning to provide a recommended system for each student.

As for the MATS prototype system, its working mechanism has been described. Three learning cases were presented to show the differences between MATS and the traditional knowledge-based ATS when making tutoring decisions. The main future work is to finalize the implementation of the behavior analysis module, to put it into real use and perform evaluations to see if considering motivational states will keep students motivated and will improve students' learning performance.

BIBLIOGRAPHY

[1] J. Keefe and J. Jenkins, *Personalized Instruction: Changing Classroom Practice*, Larchmont, New York, 2000

[2] P. Brusilovsky, "Intelligent tutoring systems for World-Wide Web," *Proc. of Third International WWW Conference (Posters),* pp.42-45, April, 1995.

[3] P. Brusilovsky, "Adaptive Hypermedia," *User Modeling and User-Adapted Interaction*, Vol. 11, No. 1-2, pp. 87-110, 2001.

[4] R. Riding and S. Rayner, *Cognitive Styles and Learning Strategies,* London: David Fulton Publishers, 2000.

[5] G. Gibbs and A. Jenkins, *Teaching large classes in higher education: how to maintain quality with reduced resources*, London, Kegan page, 1992.

[6] E. Carbone, *Teaching Large Classes: Tools and Strategies*, SAGE publications, 1998.

[7] S. Gilbert, "Quality Education: Does Class Size Matter?" *Research File,* Vol. 1, No. 1, pp. 1-7, 1995.

[8] G. Gibbs and A. Jenkins, *"Teaching Large Classes in Higher Education, London,"* Kegan Paul, 1992.

[9] G.M. Novak, E.T. Patterson, A.D., Gavrin and W. Christian, *"Just-in-time Teaching: Blending Active Learning with Web Technology,"* Prentice Hall, 1999.

[10] Y. Lian and Y.H. Tan, "An Educator-to-Learner Portal for Just-in-Time Teaching," *Proc. of the Second Symposium on Teaching and Learning in Higher Education*, pp. 420-425, 2002.

[11] Y. Lian, “Learning Digital Signal Processing at Six Degrees of Freedom,” *Proc. of the Second Symposium on Teaching and Learning in Higher Education*, pp. 444-449, 2002.

[12] Y. Lian, “Teaching large Class Effectively with the Help of IT,” *Proc. of 2002 International Seminar on Information and Communication technologies*, pp. 225-230, 2002.

[13] Y. Lian, “The Joy of Learning DSP in a Large Class,” *Proc. of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 753-756, 2003.

[14] B.S. Bloom, “The 2 Sigma problem: The search for methods of group instruction as effective as one-to-one tutoring,” *Educational Researcher,* Vol. 13, No. 6, pp. 4-16, 1984.

[15] S. Paul *et al.*, “Features and Advantages of WME: A Web-based Mathematics Education System,” *Proc. IEEE Southeast Conference,* pp.621-629, April 2005.

[16] M. Koc, “Individual Learner Differences In Web-based Learning Environments: From Cognitive, Affective and Social-cultural Perspectives,” *Turkish Online Journal of Distance Education,* Vol. 6, No. 4, 2005.

[17] R. Martens, J. Gulikers and T. Bastaens, “The impact of intrinsic motivation on e-learning in authentic computer tasks,” *Journal of Computer Assisted Learning,* Vol. 20, No. 5, pp. 368-376, 2004.

[18] D. Goleman, *“Emotional Intelligence: Why can matter more than IQ,”* London: Boomsbury, 1996.

[19] R.E. Snow, L. Corno and D.N. Jackson, “Individual differences in affective and conative functions,” *Handbook of Educational Psychology,* pp. 243-310, New York: McMillan 1996.

[20] L. Corno, “Conative individual differences in learning,” *Intelligence and Personality: Bridging the gap in theory and measurement,* pp.121-138, Mahwah: Lawrence Erlbaum, 2001.

[21] R. Kanfer and B.L. McCombs, "Motivation: Applying current theory to critical issues in training," *Training and Retraining: A handbook for business, industry, government, and the military,* pp. 85-108, New York: MacMillan, 2000.

[22] C. A. Carver, R. A. Howard and W. D. Lane, "Enhancing Student Learning Through Hypermedia Courseware and Incorporation of Student Learning Styles," *IEEE Transactions on Education*, Vol. 42, No. 1, pp. 33-38, 1999.

[23] A. Vincent and D. Ross, "Personalized Training: Determine Learning Styles, Personality Types and Multiple Intelligence Online," *The Learning Organization*, Vol. 8, No. 1, pp. 36-43, 2001.

[24] C.M. Reigeluth, "A new paradigm of ISD," *Educational Technology & Society*, Vol. 36, No. 3, pp. 13-20, 1996.

[25] R.M. Felder, "Matters of style," ASEE Prism, Vol.6, No.4, pp.18-23, 1996.

[26] K.L. Rasmussen, "Hypermedia and learning styles: Can performance be influenced?" *Journal of Multimedia and Hypermedia*, Vol. 7, No. 4, pp.291-308, 1998.

[27] D. Kolb and R. Fry, "Towards an applied theory of experiential learning," *Theories of group processes*, pp. 33-58, 1975.

[28] H. Gardner, *Multiple Intelligences: The Theory in Practice,* New York: Basic books, 1993.

[29] R. M. Felder and L. K. Silverman, "Learning Styles and Teaching Styles in Engineering Education," *Engineering Education,* Vol. 78, No. 7, pp. 674-681, 1988.

[30] R. M. Felder, "Reaching the Second Tier: Learning and Teaching Styles in College Science Education," *Journal of College Science Teaching,* Vol. 23, No. 5, pp. 286-290, 1993.

[31] C. A. Carver, R. A. Howard and E. Lavelle, "Enhancing student learning by incorporating student learning styles into adaptive hypermedia," *Proc. of 1996*

ED-MEDIA World Conf. on Educational Multimedia and Hypermedia, Boston, MA, pp. 118-123, 1996.

[32] M. Specht and R. Oppermann, "ACE - Adaptive Courseware Environment," *The New Review of Hypermedia and Multimedia*, Vol. 4, pp. 141-161, 1998.

[33] J. E. Gilbert and C. Y. Han, "Arthur: Adapting Instruction to Accommodate Learning Style," *Proc. of World Conference on the WWW and Internet, Honolulu*, *HI: Association for the Advancement of Computing in Education,* pp.433-438, 1999.

[34] J. E. Gilbert and C. Y. Han, "Adapting Instruction in Search of 'A Significant Difference'," *Journal of Network and Computing Applications*, Vol. 22, No. 3, 1999.

[35] J.E. Gilbert and C. Y. Han, "Arthur: A Personalized Instructional System," *Journal of Computing In Higher Education,* Vol. 14, No. 1, Norris Publishers, Amherst, MA, 2002.

[36] P. Paredes and P. Rodriguez, "Considering Learning Styles in Adaptive Web-based Education," *Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics en Orlando*, Florida, July 2002, 481-485.

[37] D. A. Kolb, *Experiential Learning,* Prentice-Hall, Englewood Cliffs, NJ, D.A., 1984.

[38] P. Honey and A. Mumford, "*The Manual of Learning Styles,*" Maidenhead, 1992.

[39] B. Weiner, "*Human Motivation: Metaphors, Theories, and Research*," Sage Publications Inc., Newbury Park, London, 1992.

[40] P. Mumford and A. Mumford, *The Manual of Learning Styles*, Maidenhead, London, 1992.

[41] C. Karagiannidis, D. Sampson and F. Cardinali, "Integrating Adaptive Educational Content into Different Courses and Curricula," *Educational*

Technology & Society, Special Issue in Curriculum, Instruction, Learning and the Internet, Vol. 4, No. 3, 2001.

[42] P. Brusilovsky, "The construction and application of student models in intelligent tutoring systems," *Computer and System Sciences International,* Vol. 32, No. 1, pp. 70-89, 1994.

[43] A. Granic, S. Stankov and V. Glavinic, "User interface aspects of an intelligent tutoring system, *22nd International Conference Information Technology Interfaces*, Pula, pp.369-372, June 13-16, 2000.

[44] P. Brusilovsky, "Adaptive and Intelligent Technologies for Web-based Education," *Special Issue on Intelligent Systems and Teleteaching,* Vol. 4, pp. 19-25, 1999.

[45] P. Brusilovsky, "Integrating hypermedia and intelligent tutoring technologies: from systems to authoring tools," *New media and telematic technologies for education in Eastern European countries,* pp. 129-140, 1997.

[46] S. Habeshaw, G. Gibbs, and T. Habshaw, "53 Problems with Large Classes. Making the Best of a Bad Job," Bristol, Technical & Educational Services, 1992.

[47] V. Shute and R. Glaser, "A large-scale evaluation of an intelligent discovery world: Smithtown," *Interactive Learning Environments*, Vol. 1, pp. 51-77, 1991.

[48] S. Ritter and K. R. Koedinger, "An architecture for plug-in tutoring agents," *Journal of Artificial Intelligence in Education*, Vol. 7, pp. 315-347, 1997.

[49] N.K. Person, A.C. Graesser, and L. Bautista, "Evaluating student learning gains in two versions of AutoTutor," *Proc. of AI in Education 2001 Conference*, Amsterdam, pp. 286-293, 2001.

[50] C. P. Rosé *et al.*, "Interactive conceptual tutoring in Atlas-Andes," *Proc. of Artificial Intelligence in Education 2001 Conference*, Amsterdam, pp.256-266, 2001.

[51] K. R. Koedinger *et al.*, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, Vol. 8, pp. 30-43, 1997.

[52] T. Murray and S. Blessing, *"Authoring Tools for Advanced Technology Learning Environments,"* Amsterdam: Kluwer Academic Publishers, 2003.

[53] B.P. Woolf and P. Cunningham, "Building a community memory for intelligent tutoring systems," *Proc. Of Association for the Advancement of Artificial Intelligence 1987,* pp. 82-89, 1987.

[54] K. L. McGraw, "Performance Support Systems: Integrating AI, Hypermedia and CBT to enhance user performance," *International Journal of Artificial Intelligence in Education,* Vol. 5, No. 1, pp. 3-26, 1994.

[55] P. Brusilovsky, "Adaptive Hypermedia," *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue, Vol. 11, pp. 87-110, 2001.

[56] P. Brusilovsky, "Adaptive and Intelligent Technologies for Web-based Education," *Special Issue on Intelligent Systems and Teleteaching*, Vol. 4, pp. 19-25, 1999.

[57] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia," *User Modeling and User Adapted Interaction*, Vol. 6, No. 2-3, pp. 87-129,1996.

[58] N. Hammond, "Hypermedia and learning: Who guides whom?" *Proc. of the 2nd International Conference of Computer Assisted Learning*, pp. 67-181, 1989.

[59] M. Koc, "Individual learner differences in web-based learning environments: From cognitive, affective and social-cultural perspectives," *Turkish Online Journal of Distance Education*, Vol. 6, No. 4, Retrieved November 3, 2005, from http://tojde.anadolu.edu.tr/index.htm.

[60] D. A. Kolb, *Experiential learning: Experience as the source of learning and development*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[61] P. Honey and A. Mumford, *The manual of learning styles*, Peter Honey, Maidenhead, 1982.

[62] N. Fleming and C.C. Bonwell, "VARK (visual aural read/write kinesthetic): A guide to learning styles," www.vark-learn.com, Retrieved 03/11/02 from source. Christchurch, New Zealand: Lincoln University, 1998.

[63] P. Paredes and P. Rodriguez, "Considering learning styles in adaptive web-based education," *Sixth World Multiconference on Systemics, Cybernetics and Informatics,* pp. 481-485, 2002.

[64] P. García *et al.,* "Using Bayesian Networks to Detect Students Learning Styles in a Web-based Education System," *Proc of Argentine Symposium on Artificial Intelligence*, Rosario, pp.115-126, 2005.

[65] J.J. Lo and P. C. Shu, "Identification of learning styles online by observing learners' browsing behavior through a neural network," *British Journal of Educational Technology*, Vol. 36, No. 1, pp. 43-55, 2005.

[66] R. M. Felder and J. Spurlin, "Applications, Reliability and Validity of the Index of Learning Styles," *International Journal on Engineering Education*, Vol. 21, pp. 103-112, 2005.

[67] J. Hayes and C.W. Allinson, "The Implications of Learning Styles for Training and Development: A Discussion of the Matching Hypothesis," *British Journal of Management*, Vol. 7, No. 1, pp.63-73, 1996.

[68] T. J. Shuell, "Cognitive Conceptions of Learning," *Review of Educational Research*, Vol. 56, No. 4, pp. 411-436, 1986.

[69] N. Ford N, "Levels and types of mediation in instructional systems: an individual differences approach," *International journal of Human-Computer studies*, Vol. 43, pp. 241-259, 1995.

[70] M. Richard and K. Silverman, "Learning and Teaching Styles in Engineering Education," *Engineering Education*, Vol. 78, No. 7, pp. 674-681, 1988.

[71] N. Ford and Y. Sherry Chen, "Matching/mismatching revisited: an empirical study of learning and teaching styles," *British Journal of Educational Technology*, Vol. 32, No. 1, pp. 5-22, 2001.

[72] D. Robotham, "*The application of learning style theory in higher education teaching*," Retrieved from http://www.chelt.ac.uk/gdn/discuss/kolb2.htm, 2002.

[73] D. Goleman, "*Emotional intelligence: Why it can Matter more than IQ*," Boomsbury London. First published by Bantam Books, New York, 1995.

[74] M. Lepper *et al.*, "Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors," *Computers as Cognitive Tools*, pp. 75-105, 1993.

[75] Y.P. Hu and Y. Lian, "An Open Learning Environment for Large Classroom Teaching," *International Conference IEEE ICL'2006 Interactive Computer Aided Learning*, Villach, Austria, 27-29 September 2006.

[76] D. Vicente and H. Pain, "Motivation diagnosis in intelligent tutoring systems," in *Proc. of the Fourth International Conference on Intelligent Tutoring Systems*, pp. 86-95, San Antonio, Texas, Usa,1998.

[77] Y. Matsubara and M. Nagamachi, "Motivation systems and motivation models for intelligent tutoring," *Proc. of the Third International Conference in Intelligent Tutoring Systems*, pp. 139-147, 1996.

[78] D. Whitelock and E. Scanlon, "Motivation, media and motion: Reviewing a computer supported collaborative learning experience," *Proc. of the European Conference on Artificial Intelligence in Education*, 1996.

[79] D. Soldato, "Motivation in tutoring systems," Tech. Rep. CSRP 303, School of Cognitive and Computing Sciences, The University of Sussex, UK, 1994.

[80] D. V. Angel and H. Pain, "Motivation self-report in ITS," *Proc. of the Ninth World Conference on Artificial Intelligence in Education*, pp. 648-650, Amsterdam, 1999.

[81] W. Hioe and D. J. Campbell, "An expert system for the diagnosis of motivation-related job performance problems: Initial efforts," DICS publications no. TRA3/88, Department of Information Systems and Computer Science, National University of Singapore. 1988.

[82] A. Kapoor, R.W. Picard and Y. Ivanov, "Probabilistic Combination of Multiple Modalities to Detect Interest," *International Conference on Pattern Recognition,* Cambridge, U.K., 2004.

[83] S. K. D'Mello *et al.*, "Integrating Affect Sensors in an Intelligent Tutoring System," *Affective Interactions: The Computer in the Affective Loop Workshop at 2005 International conference on Intelligent User Interfaces*, pp. 7-13, New York: AMC Press, 2005.

[84] R. Fernandez and R. W. Picard, "Classical and Novel Discriminant Features for Affect Recognition from Speech," *Interspeech 2005 - Eurospeech 9th European Conference on Speech Communication and Technology*, Lisbon Portugal, September 4-8, 2005.

[85] K. Cotton, "Monitoring student learning in the classroom," School Improvement Research Series (SIRS). Northwest Regional Educational Laboratory, U.S. Department of Education., 1988. http://www.nwrel.org/scpd/sirs/2/cu4.html.

[86] G. Weber and M. Specht, "User modeling and Adaptive Navigation Support in WWW Based Tutoring Systems," *Proc. of the 6th International User Modeling conference*, pp. 289-300, 1997.

[87] J. Eklund, P. Brusilovsky and E. Schwarz, „Adaptive Textbooks on the WWW," *Proc. The Third Australian Conference on the World Wide Web*, pp. 186-192, 1997.

[88] R. M. Baecker *et al.*, "From Customizable Systems to Intelligent Agents," *Human-Computer Interaction: Toward the Year 2000*, Chapter 12, pp. 783-791, 1995.

[89] K. Papanikolaou and M. Grigoriadou, "Accommodating learning style characteristics in adaptive educational hypermedia," *Workshop on individual differences in adaptive hypermedia organized in AH2004*, Part I, pp. 77-86, Eindhoven, 22-26 August, 2004.

[90] T. W. Yan *et al.*, "From User access patterns to dynamic hypertext linking," *Computer Networks*, Vol. 28, pp. 7-11, 1996.

[91] A. Jameson, "Numerical uncertainty management in user and student modeling: An overview of systems and issues," *User Modeling and User-Adapted Interaction*, Vol. 5, pp. 193-251, 1996.

[92] C. L. Posey and L. W. Hawkes, "Neural networks applied to knowledge acquisition in the student model," *Information Sciences*, Vol. 88, pp. 275-298, 1996.

[93] J. E. Beck and B. P. Woolf, "Using a learning agent with a student model," *Proc. of the fourth International Conference on Intelligent Tutoring System1998*, pp. 6-15, 1998.

[94] R. Stathacopoulou, G.D. Magoulas and M. Grigoriadou, "Neural network-based fuzzy modeling of the student in intelligent tutoring systems," *Proc. of International Joint Conference on Neural Networks*, Washington, 1999.

[95] Z. Q. Liu and F. Yan, "Fuzzy neural network in case-based diagnostic system," *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 2, pp. 209-222, 1997.

[96] V. J. Shute and J. Psotka, "Intelligent Tutoring Systems: Past, Present and Future," *Handbook of Research on Educational Communications and Technology*: *Scholastic Publications*, pp. 1-99, 1996.

[97] R. J. Mislevy, "Evidence and inference in educational assessment," *Psychometrika*, Vol. 59, pp. 439-483, 1994.

[98] J. Pearl, *Probabilistic Reasoning in Expert Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann Publishers, Inc, 1988.

[99] C. Conati *et al.*, "On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks," *Proc. of the Seventh International Conference on User Modeling*, Springer-Verlag, pp. 231-242, 1997.

[100] A. Gertner and K. VanLehn, "Andes: A Coached Problem Solving Environment for Physics," *Proc. of Fifth International Conference on Intelligent Tutoring Systems*, Springer-Verlag, pp. 133-142, 2000.

[101] C. Conati and K. VanLehn, "POLA: a student modeling framework for Probabilistic On-Line Assessment of problem solving performance," *Proc. of the Fifth International Conference on User Modeling*, pp. 75-82.

[102] T. Murray, "Authoring Intelligent Tutoring Systems: An analysis of the state of the art," *International Journal of Artificial Intelligence in Education*, Vol. 10, pp. 98-129, 1999.

[103] P. Brusilovsky, "Developing Adaptive Educational Hypermedia Systems: from Design Models to Authoring Tools," *Authoring Tools for Advanced Technology Learning Environments*, pp. 377 – 409, 2003.

[104] S. Ritter and S. Blessing, "Authoring tools for component-based learning environments," *Journal of the Learning Sciences*, Vol. 7, pp. 107-132, 1996.

[105] M. W. Goldberg, S. Salari and P. Swoboda, "World Wide Web - course tool: An environment for building WWW-based courses," *Journal of Computer Networks and ISDN Systems*, Vol. 28, pp. 1219-1231, 1996.

[106] N. Major, S. Ainsworth and D. Wood, "REDEEM: Exploiting Symbiosis between Psychology and Authoring Environments," *International Journal of Artificial Intelligence in Education*, Vol. 8, pp. 317-340, 1997.

[107] P. Brusilovsky, J. Eklund and E. Schwarz, "Web-based education for all: A tool for developing adaptive courseware," *Journal of Computer Networks and ISDN Systems*, Vol. 30, pp. 291-300, 1998.

[108] M. Virvou and M. Moundridou, "A Web-based authoring tool for Algebra-related ITSs," *Journal of International Forum of Educational Technology & Society and IEEE Learning Technology Task Force*, Vol. 3, pp. 61-70, 2000.

[109] A. Blandford, H. Stelmaszewska and N. Bryan-Kinns, "Use of multiple digital libraries: A case study," *Proc. of the Joint Conference on Digital Libraries*, New York: ACM Press, 2001.

[110] G. Marchionini, C. Plaisant and A. Komlodi, "Interfaces and tools for the Library of Congress National Digital Library Program," *Information Processing & Management*, Vol. 34, No. 5, pp. 535-555, 1998.

[111] K. Papanikolaou, M. Grigoriadou and H. Kornilakis, "Instructional and interface design in an adaptive educational hypermedia system," *Panhellenic Conference in Human-Computer Interaction (PC-HCI 2001)*, Patra, Greece, 2001.

[112] G. McCalla and J. E. Greer, "Granularity-Based Reasoning and Belief Revision in Student Models in Student Models: The Key to Individualized Educational Systems," New York: Springer Verlag, 1994.

[113] K. A. Papanikolaou *et al.*, "Personalizing the interaction in a web-based educational hypermedia system: the case of INSPIRE," *User Modeling and User-Adapted Interact*. Vol. 13, pp. 213-267, 2003.

[114] B. Carr and I. P. Goldstein, "Overlays: a Theory of Modeling for Computer Aided Instruction," *International Journal of Man-Machine Studies*, Vol. 5, pp. 215-236, 1977.

[115] P. Holt *et al.*, "The State of Student Modeling," *Student Modeling: The Key to Individualized Knowledge-Based Instruction*, pp. 3-35, Springer-Verlag, 1991.

[116] C. Conati *et al.*, "On-line student modeling for coached problem solving using Bayesian networks," *International Conference on User modeling*, Chia Laguna, Italy, pp. 231-242, 1997.

[117] E. Horvitz *et al.,* "The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users," *Proc. of the 14th Conference Uncertainty in AI, Morgan Kaufmann*, San Francisco, USA, pp. 256-265, 1998.

[118] C. Conati, A. Gertner and K. Vanlehn, "Using Bayesian networks to manage uncertainty in student modeling," *Journal of User Modeling and User-Adapted Interaction*, Vol. 12, No. 4, pp. 371-417, 2002.

[119] N. Major, "How Generic Can Authoring Shells Become?" *Workshop on Authoring Shells for Intelligent Tutoring Systems,* pp. 69-72, 1995.

[120] E. Millán, J. Pérez-de-la-Cruz and E. Suárez, "Adaptive Bayesian Networks for Multilevel Student Modeling," *Proc. of 5th International Conference on Intelligent Tutoring Systems*, Springer-Verlag, pp. 534- 543, 2000.

[121] E. Millian *et al.*, "Using Bayesian networks in computerized adaptive tests," *Computers and Education in the 21st Century*, Kluwer, pp. 217-228, 2000.

[122] T. M. Mitchell, "Bayesian Learning in Machine learning," New York; London: McGraw-Hill, pp. 154-200, 1997.

[123] S. J. Russell and P. Norvig, "Statistical Learning Methods," *Artificial intelligence: a modern approach*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 2003.

[124] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 2003.

[125] T. M. Mitchell, *Machine learning*, New York; London: McGraw-Hill, 1997.

[126] R. Riding and G. Douglas, "The effect of cognitive style and mode of presentation on learning performance," *British Journal of Educational Psychology*, Vol.63, pp.297-307, 1993.

[127] C. McLoughlin, "The implications of the research literature on learning styles for the design of instructional material," *Australian Journal of Educational Technology*, Vol. 15, No. 3, pp. 222-241, 1999.

[128] J. Vomlel, "Two applications of Bayesian networks," Available "citeseer.ist.psu.edu/559264.html".

[129] L. M.Rudner, "An examination of decision-theory adaptive testing procedures," *American education research association*, 2002.

[130] M. C. Desmarais, A. Maluf and J. Liu, "User-expertise modeling with empirically derived probabilistic implication networks," *User modeling and user-adapted interaction*, Vol. 5, pp. 283-315, 1996.

[131] E. H. Mory, "The Use of Informational Feedback in Instruction: Implications for Future Research," *Educational Technology Research and Development*, Vol. 40, No. 3, pp. 5-20, 1992.

[132] K. VanLehn, S. Ohlsson and R. Nason, "Applications of Simulated Students: An Exploration," *Journal of Artificial Intelligence in Education*, Vol. 5, No. 2, pp. 135-175, 1994.

[133] J.S. Mertz, "Using a simulated student for instructional design," *International Journal of Artificial Intelligence in Education*, Vol. 8, pp. 16-141, 1997.

[134] C. Crouch and E. Mazur, "Peer instruction: Ten years of experience and results," *American Association of Physics Teachers*, Vol. 69, No. 9, pp. 970-977, 2001.

[135] P. Albacete and K. VanLehn, "Evaluating the Effectiveness of a Cognitive Tutor for Fundamental Physics Concepts," *Proc. of the 22nd Annual Meeting of the Cognitive Science Society*, Erlbaum Associates: Mahwah, pp. 25-30, 2000.

[136] J. Cohen, *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Erlbaum, 1998.

[137] P.L. Albacete and K. VanLehn, "The Conceptual Helper: an Intelligent Tutoring System for Teaching Fundamental Physics Concepts," *Intelligent Tutoring Systems: 5th International Conference*, pp. 564-573, 2000.

[138] A. Mitrovic, B. Martin, and M. Mayo, "Using evaluation to shape ITS design: Results and Experiences with SQL-Tutor," *User Modeling and User-Adapted Interaction*, Vol. 12, No. 2-3, pp. 243-279, 2002.

[139] J. R. Anderson *et al.*, "Cognitive tutors: Lessons learned," *Journal of the Learning sciences*, Vol. 4, No. 2, pp. 167-207, 1995.

[140] D. Vicente and H. Pain, “Informing the detection of the students' motivational state: An empirical study,” *Proc. of the International Conference on Intelligent Tutoring Systems*, pp. 933-943, Biarritz, France, 2002.

[141] D. Vicente, “Towards tutoring systems that detect students' motivation: an investigation,” Ph.D. thesis, School of Informatics, University of Edinburgh, UK, 2003.

[142] T. D. Soldato and B. D. Boulay, “Implementation of motivational tactics in tutoring systems,” *International Journal of Artificial Intelligence in Education*, Vol. 6, pp. 337-378, 1995.

[143] T. Dean and K. Kanazawa, “Probabilistic temporal reasoning,” *Proc. of the 7th national conference on artificial intelligence*, pp. 524-529, 1988.

[144] M. I. Jordan. An introduction to probabilistic graphical models. Book draft, June 2003.

[145] K. P. Murphy, “Dynamic Bayesian networks: representation, inference and learning,” PhD thesis, Computer science division, University of California, Berkeley, CA, USA, 2002.

[146] P. Magni, “A new approach to optimal dynamic therapy planning,” in Transforming Health Care through Informatics: Proceedings of the 1999 AMIA Annual Symposium, 1999.

[147] R. C. Murray and K. VanLehn, “DT Tutor: A Decision-Theoretic, Dynamic Approach for Optimal Selection of Tutorial Actions,” *Intelligent Tutoring Systems, 5th International Conference,* pp. 153-162. New York: Springer, 2000.

[148] T. Dean and M. Welman, *Planning and Control, San Mateo*, CA: Morgan Kaufmann Publishers, 1991.

[149] E. Millán and J. L. Pérez-de-la-Cruz, "A Bayesian diagnostic algorithm for student modeling and its evaluation," *User Modeling and User-Adapted Interaction*, Vol. 12, pp. 281-230, 2002.

[150] K. VanLehn, Z. Niu, S. Siler and A. S. Gertner, "Student modeling from conventional test data: a Bayesian approach without priors," *Proceedings of the Intelligent Tutoring Systems* , pp. 434-443, 1998.

[151] S. Russell and P. Norvig, *Artificial Intelligence: A modern approach*. Upper Saddle River: Prentice Hall, 1995.

[152] C. Carmona, E. Millán, J. L. Pérez-de-la-Cruz, M. Trella, and R. Conejo, "Introducing prerequisite relations in a multi-layered bayesian student model," Proceedings of User Modeling, pp. 347-356, 2005.

[153] J. A. Collins, J. E. Greer, and S. H. Huang, "Adaptive assessment using granularity hierarchies and Bayesian nets," *Proceedings of the Intelligent Tutoring Systems*, pp. 569-577, 1996.

[154] S. Ur and K. VanLehn, "STEPS: A Simulated, Tutorable Physics Student," *Journal of Artificial Intelligence and Education*, Vol. 6, pp.405-437, 1995

[155] T. W. Chan and A. B. Baskin, "Learning Companion Systems," *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education*, 1990.

[156] T. W. Chan and C. Y. Chou, "Simulating a Learning Companion in Reciprocal Tutoring Systems," *Proceedings of Computer Support for Collaborative Learning*, Indiana University, Bloomington, IN, 1995.

[157] K. VanLehn and Z. Niu, "Bayesian student modeling, user interfaces and feedback: A sensitivity analysis," *International Journal of Artificial Intelligence in Education*, Vol. 12, pp. 154-184, 2001.

[158] N. D. Chin, "Empirical Evaluation of User Models and User-Adapted Systems," *User Modeling and User-Adapted Interaction*, Vol.11, pp.181-194, 2001.

[159] P. Brusilovsky, C. Karagiannidis and D. Sampson, “Layered evaluation of adaptive learning systems,”, *International Journal of Continuing Engineering Education and Lifelong Learning,* Vol. 14, pp. 402 – 421, 2004.